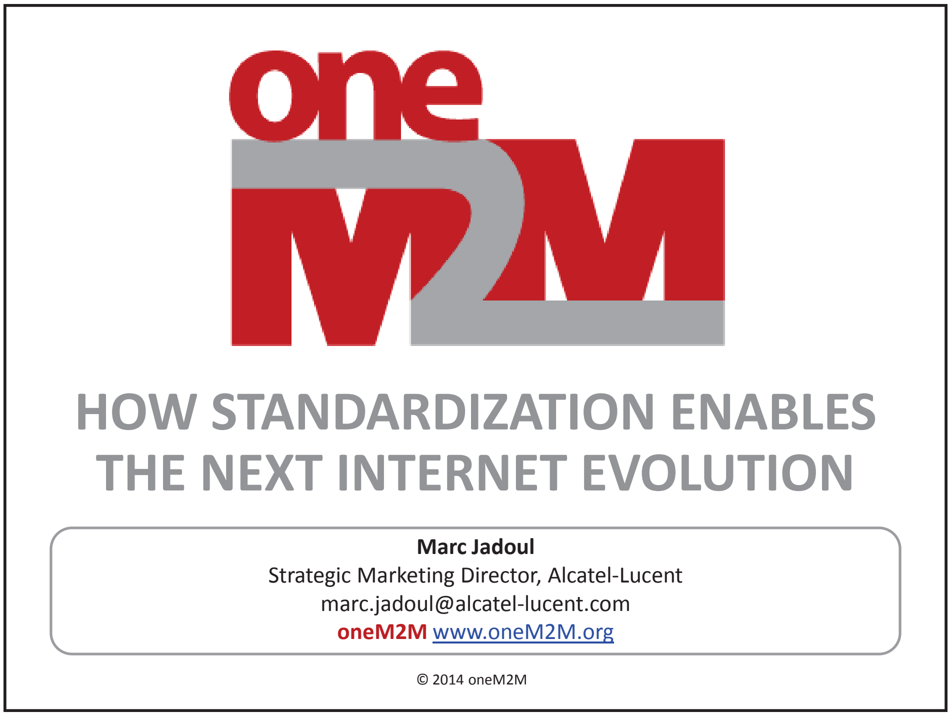
EXHIBIT 1 TO FELDMAN REPLY DECLARATION

DOCUMENTS 69 THROUGH 83 AND 125

PUBLIC VERSION

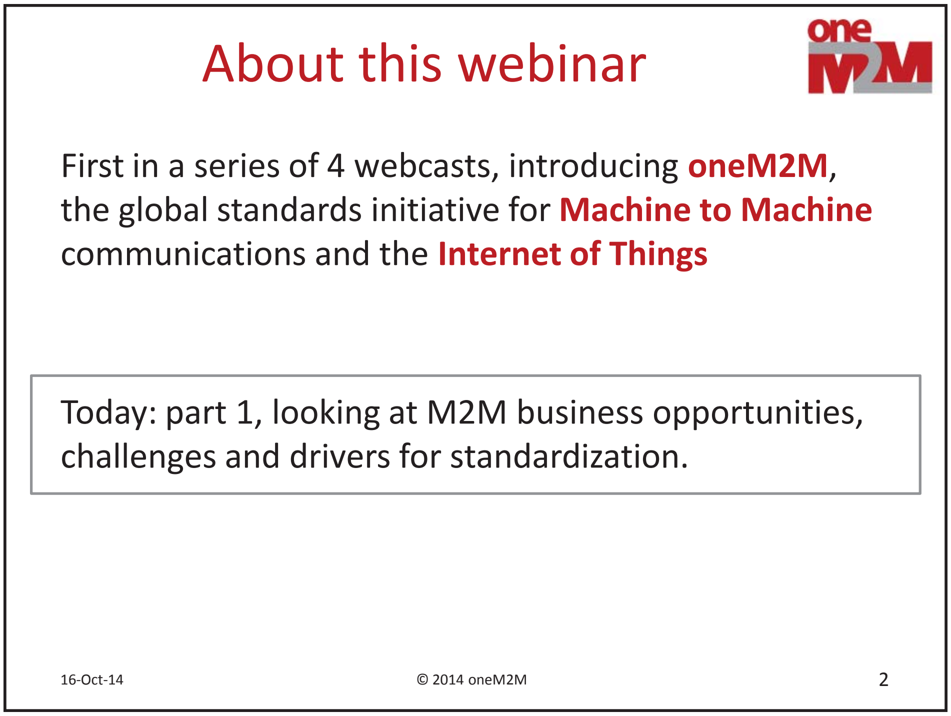
EXHIBIT 1 TO FELDMAN REPLY DECLARATION

DOCUMENT 69



HOW STANDARDIZATION ENABLES THE NEXT INTERNET EVOLUTION

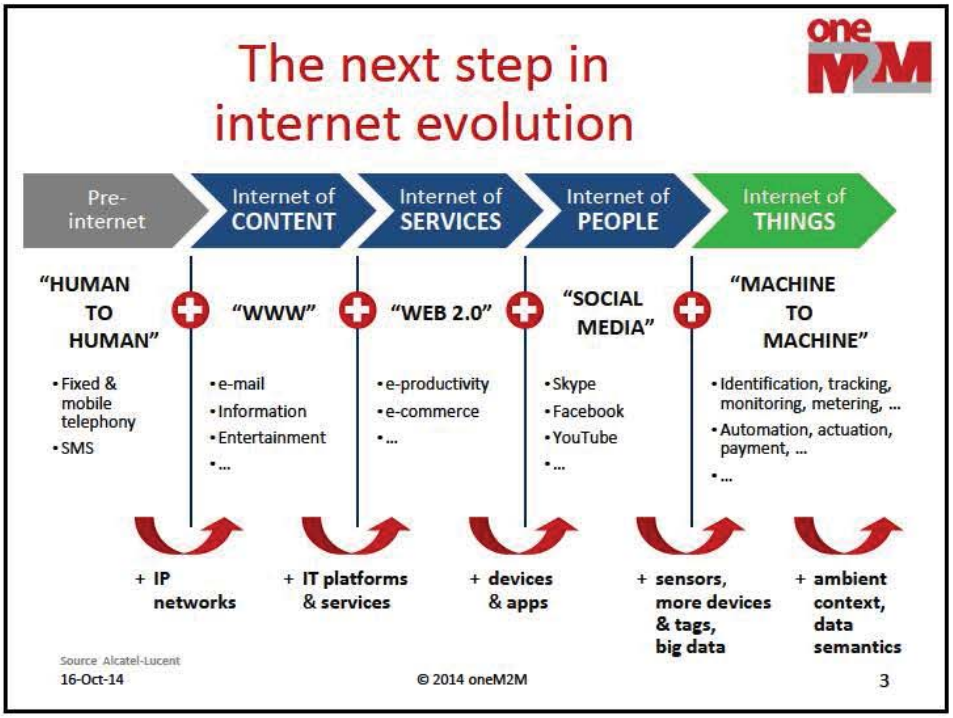
Marc Jadoul
Strategic Marketing Director, Alcatel-Lucent
marc.jadoul@alcatel-lucent.com
oneM2M www.oneM2M.org



About this webinar

First in a series of 4 webcasts, introducing **oneM2M**, the global standards initiative for **Machine to Machine** communications and the **Internet of Things**

Today: part 1, looking at M2M business opportunities, challenges and drivers for standardization.



The next step in internet evolution

Pre-internet	Internet of CONTENT	Internet of SERVICES	Internet of PEOPLE	Internet of THINGS
"HUMAN TO HUMAN"	"WWW"	"WEB 2.0"	"SOCIAL MEDIA"	"MACHINE TO MACHINE"
• Fixed & mobile telephony • SMS	• e-mail • Information • Entertainment • …	• e-productivity • e-commerce • …	• Skype • Facebook • YouTube • …	• Identification, tracking, monitoring, metering, … • Automation, actuation, payment, … • …

+ IP networks

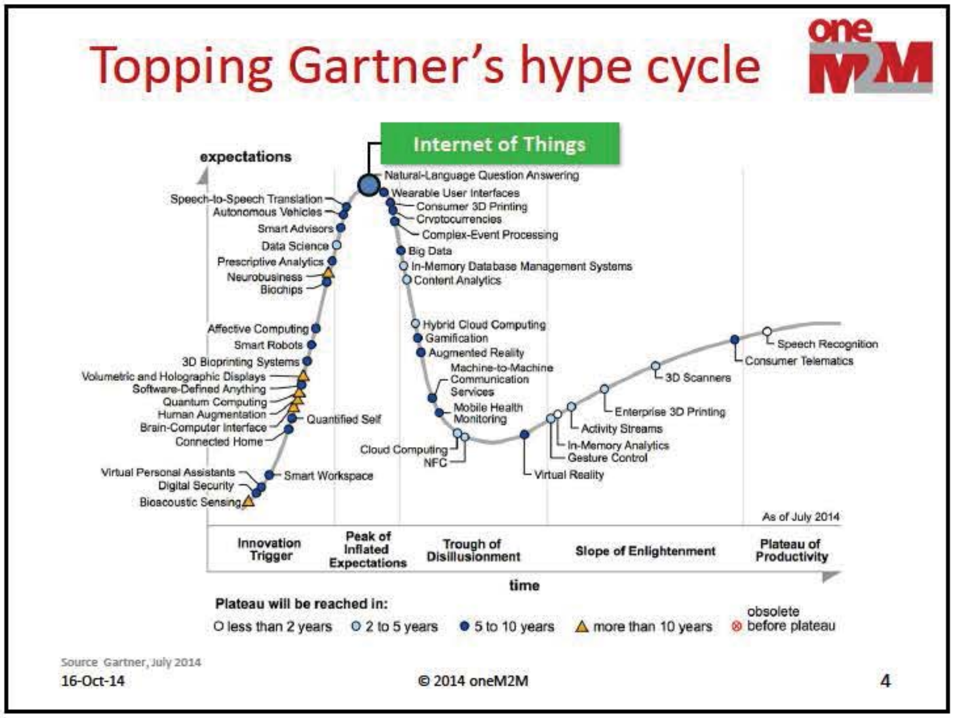
+ IT platforms & services

+ devices & apps

+ sensors, more devices & tags, big data

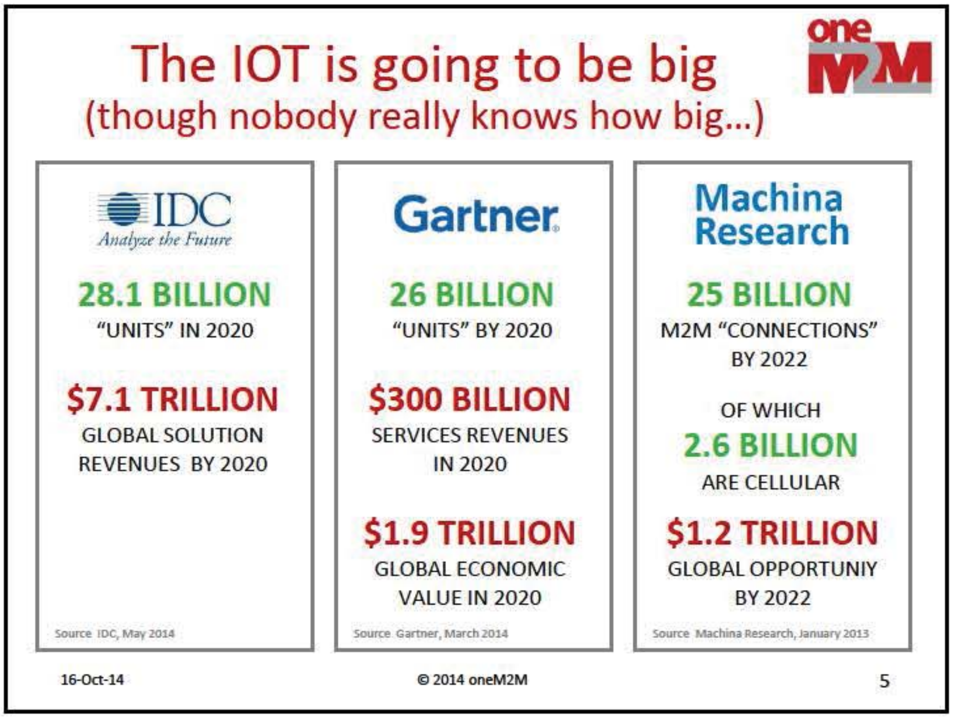
+ ambient context, data semantics

Source Alcatel-Lucent



Topping Gartner's hype cycle

Source Gartner, July 2014



The IOT is going to be big
(though nobody really knows how big...)

IDC
Analyze the Future

28.1 BILLION
"UNITS" IN 2020

$7.1 TRILLION
GLOBAL SOLUTION REVENUES BY 2020

Source IDC, May 2014

Gartner

26 BILLION
"UNITS" BY 2020

$300 BILLION
SERVICES REVENUES IN 2020

$1.9 TRILLION
GLOBAL ECONOMIC VALUE IN 2020

Source Gartner, March 2014

Machina Research

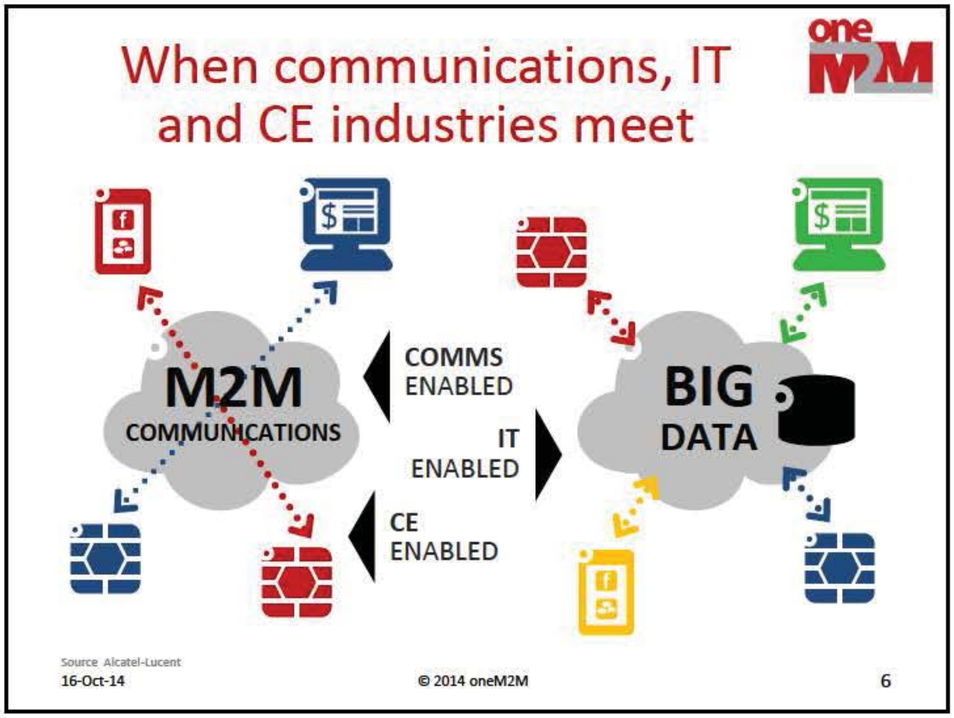
25 BILLION
M2M "CONNECTIONS" BY 2022

OF WHICH
2.6 BILLION
ARE CELLULAR

$1.2 TRILLION
GLOBAL OPPORTUNIY BY 2022

Source Machina Research, January 2013

16-Oct-14 © 2014 oneM2M 5



When communications, IT and CE industries meet

M2M COMMUNICATIONS

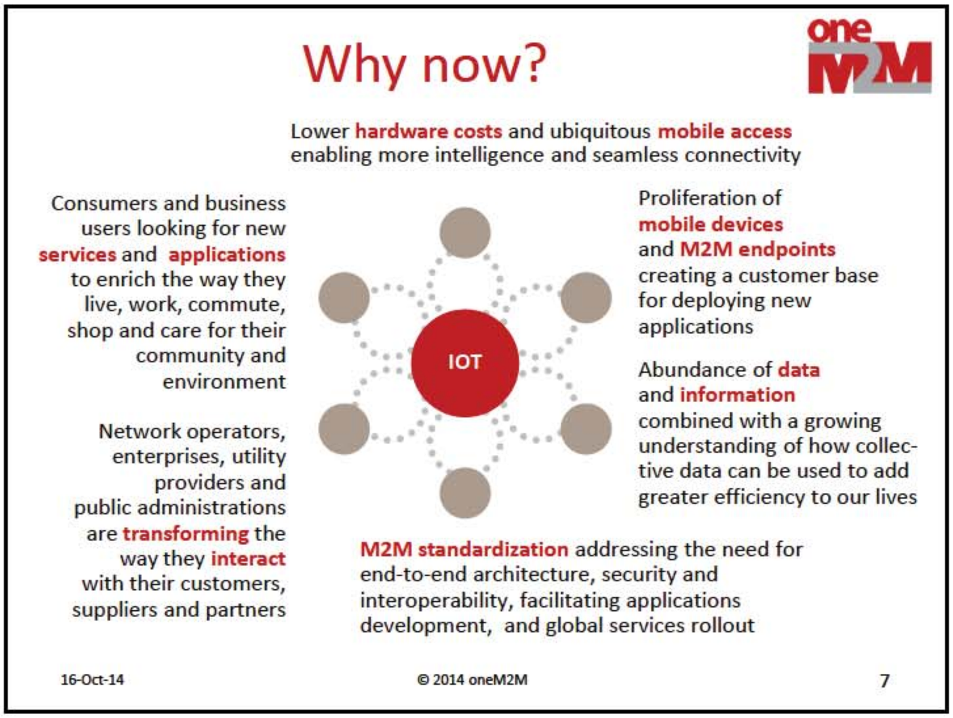
COMMS ENABLED

IT ENABLED

CE ENABLED

BIG DATA

Source Alcatel-Lucent
16-Oct-14 © 2014 oneM2M 6



Why now?

Lower **hardware costs** and ubiquitous **mobile access** enabling more intelligence and seamless connectivity

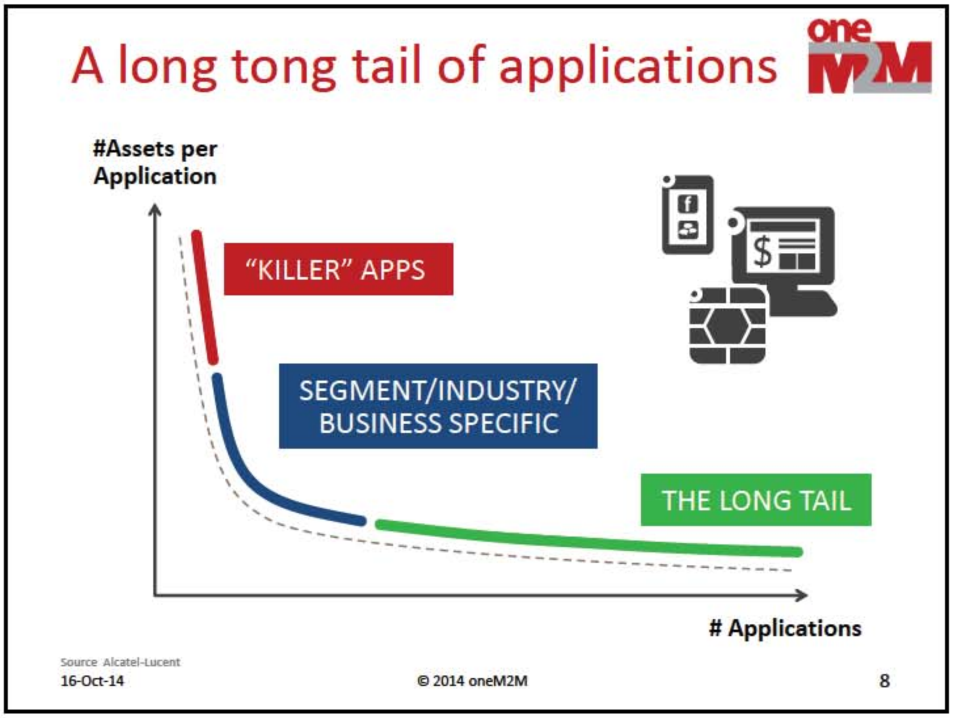
Consumers and business users looking for new **services** and **applications** to enrich the way they live, work, commute, shop and care for their community and environment

Network operators, enterprises, utility providers and public administrations are **transforming** the way they **interact** with their customers, suppliers and partners

Proliferation of **mobile devices** and **M2M endpoints** creating a customer base for deploying new applications

Abundance of **data** and **information** combined with a growing understanding of how collective data can be used to add greater efficiency to our lives

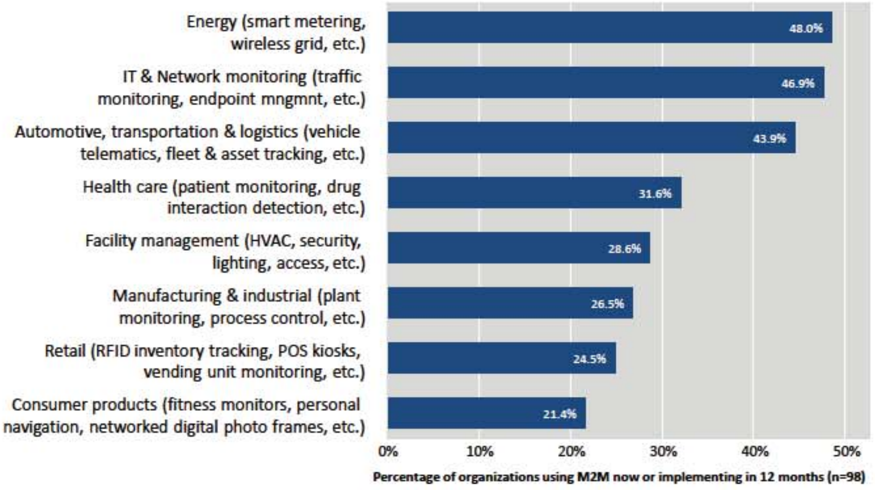
M2M standardization addressing the need for end-to-end architecture, security and interoperability, facilitating applications development, and global services rollout



A long tong tail of applications

Source Alcatel-Lucent

Where is M2M used today?

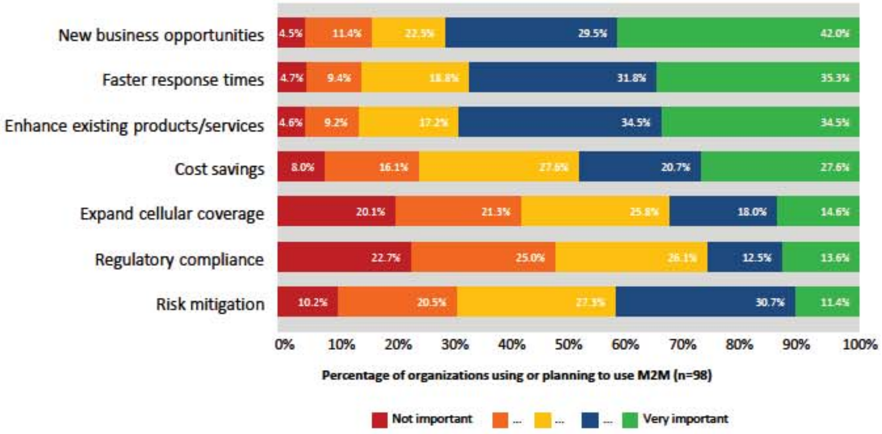


Source TechRepublic & ZDnet, 2013

16-Oct-14

© 2014 oneM2M

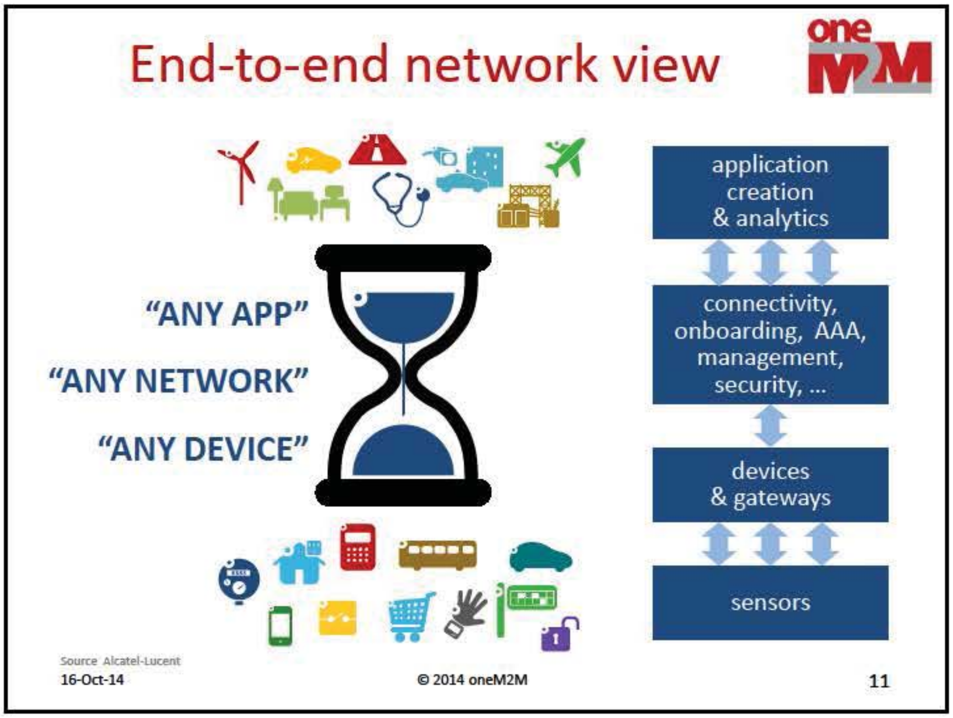
Businesses say it's about developing new opportunities



Source TechRepublic & ZDnet, 2013

16-Oct-14

© 2014 oneM2M



End-to-end network view

application creation & analytics

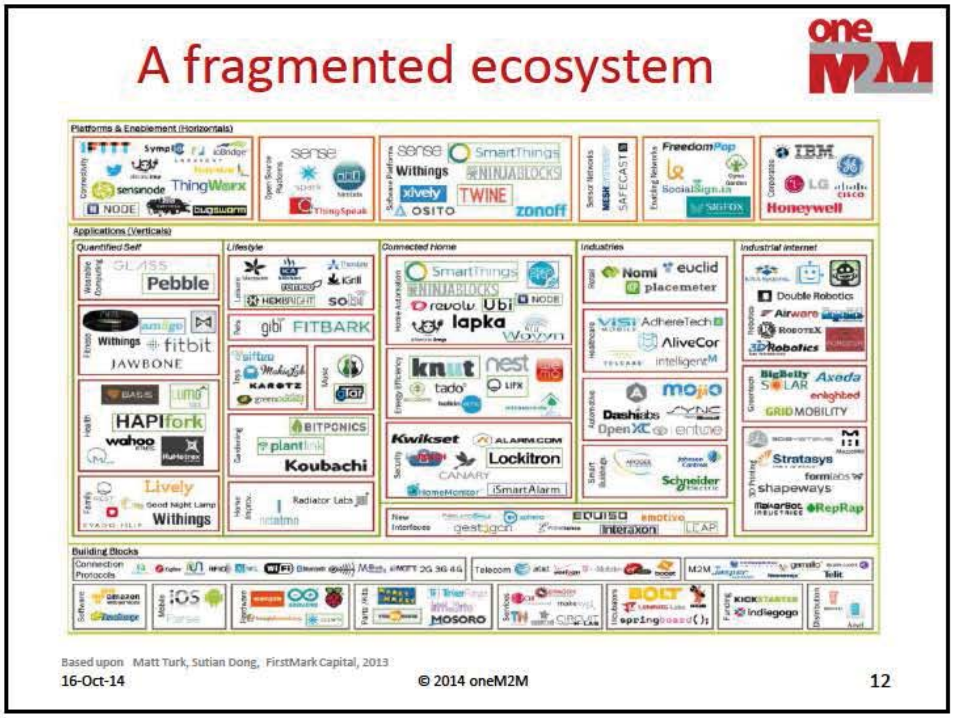
connectivity, onboarding, AAA, management, security, ...

devices & gateways

sensors

Source Alcatel-Lucent

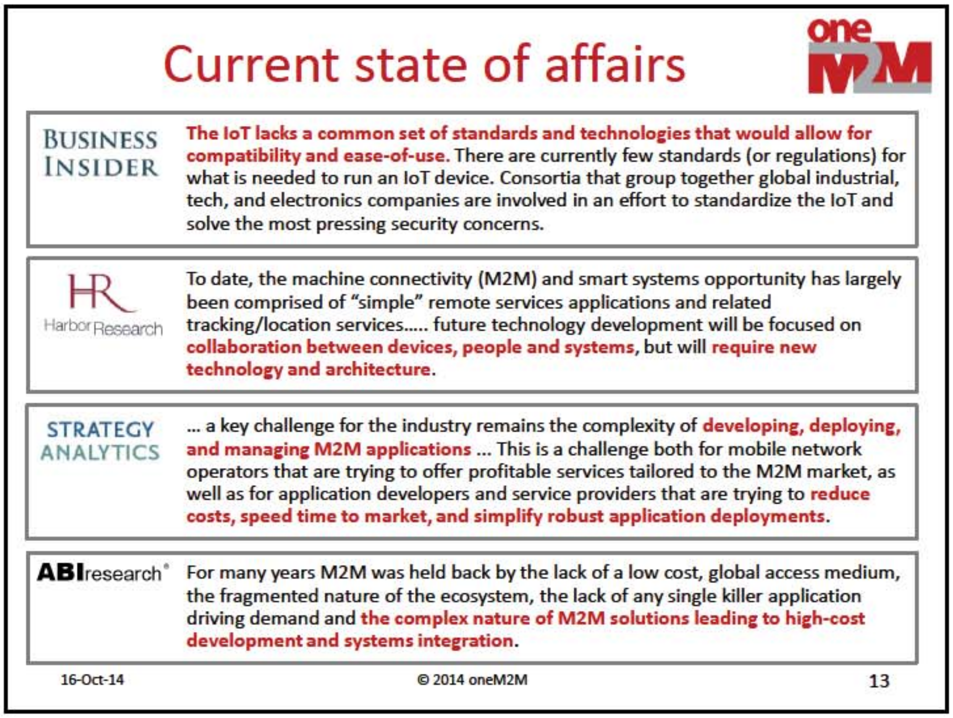
16-Oct-14 © 2014 oneM2M 11



A fragmented ecosystem

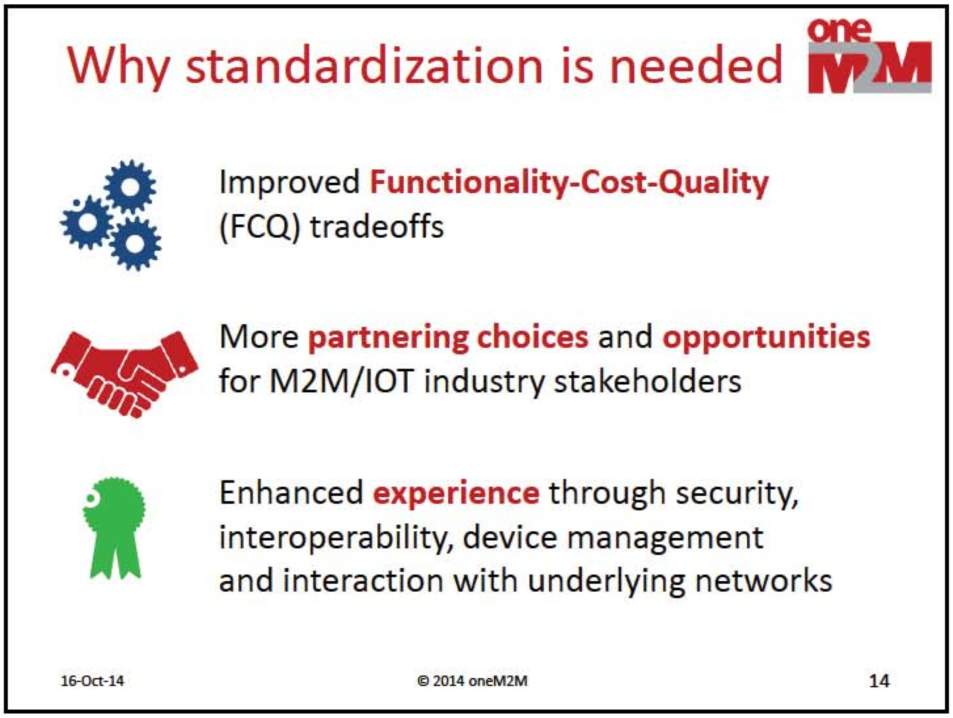
Based upon Matt Turk, Sutian Dong, FirstMark Capital, 2013

16-Oct-14 © 2014 oneM2M 12



Current state of affairs

BUSINESS INSIDER	**The IoT lacks a common set of standards and technologies that would allow for compatibility and ease-of-use.** There are currently few standards (or regulations) for what is needed to run an IoT device. Consortia that group together global industrial, tech, and electronics companies are involved in an effort to standardize the IoT and solve the most pressing security concerns.
HR Harbor Research	To date, the machine connectivity (M2M) and smart systems opportunity has largely been comprised of "simple" remote services applications and related tracking/location services..... future technology development will be focused on **collaboration between devices, people and systems**, but will **require new technology and architecture**.
STRATEGY ANALYTICS	... a key challenge for the industry remains the complexity of **developing, deploying, and managing M2M applications** ... This is a challenge both for mobile network operators that are trying to offer profitable services tailored to the M2M market, as well as for application developers and service providers that are trying to **reduce costs, speed time to market, and simplify robust application deployments**.
ABIresearch®	For many years M2M was held back by the lack of a low cost, global access medium, the fragmented nature of the ecosystem, the lack of any single killer application driving demand and **the complex nature of M2M solutions leading to high-cost development and systems integration**.

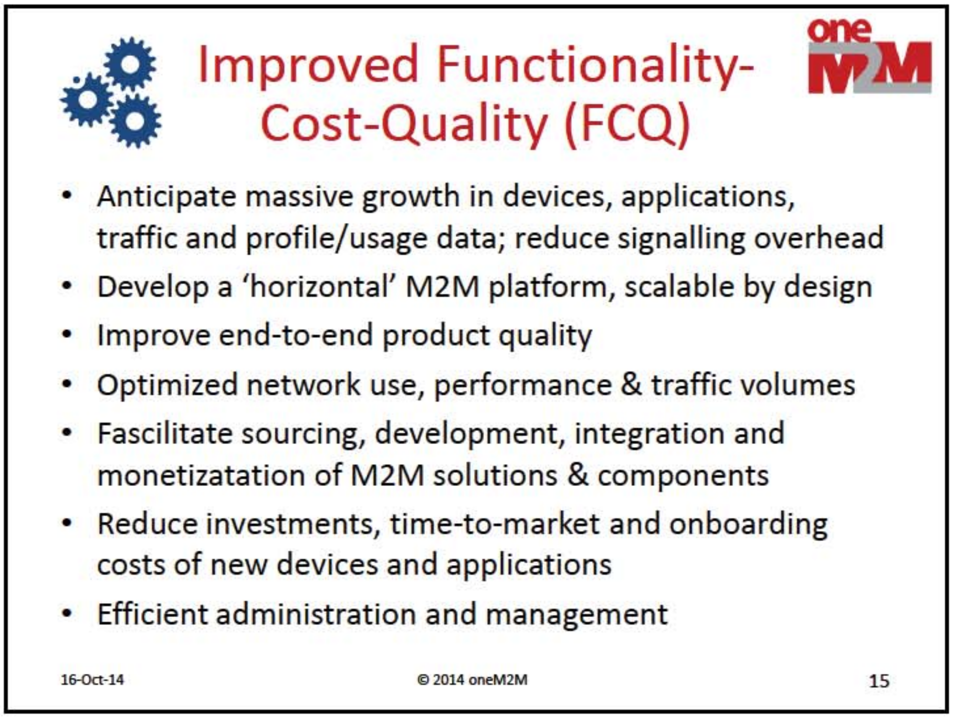


Why standardization is needed

Improved **Functionality-Cost-Quality** (FCQ) tradeoffs

More **partnering choices** and **opportunities** for M2M/IOT industry stakeholders

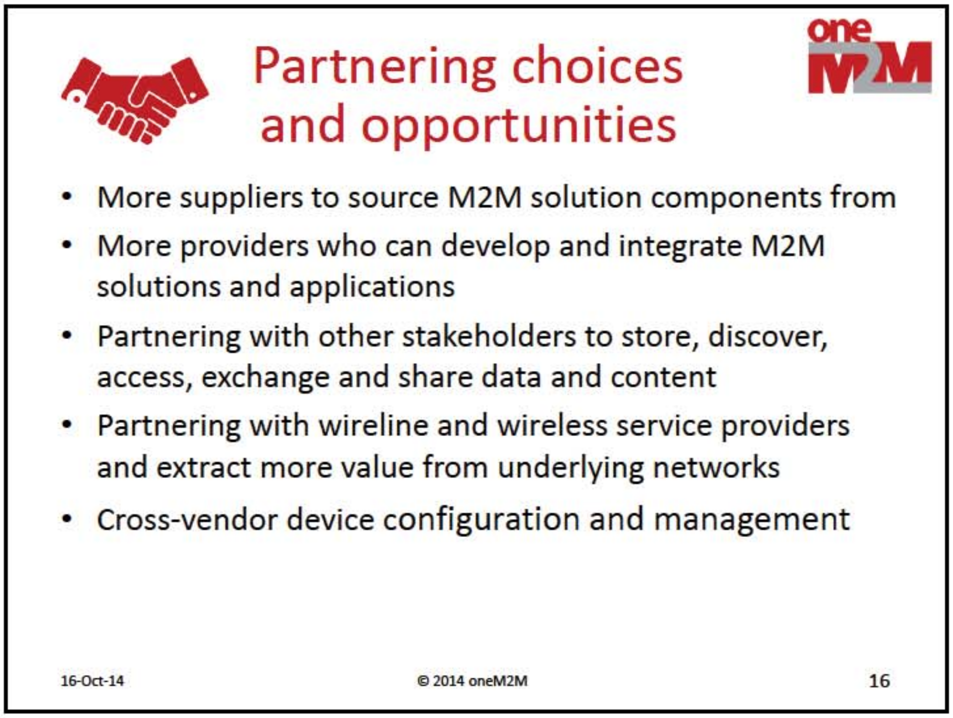
Enhanced **experience** through security, interoperability, device management and interaction with underlying networks



Improved Functionality-Cost-Quality (FCQ)

- Anticipate massive growth in devices, applications, traffic and profile/usage data; reduce signalling overhead
- Develop a ‘horizontal’ M2M platform, scalable by design
- Improve end-to-end product quality
- Optimized network use, performance & traffic volumes
- Fascilitate sourcing, development, integration and monetizatation of M2M solutions & components
- Reduce investments, time-to-market and onboarding costs of new devices and applications
- Efficient administration and management

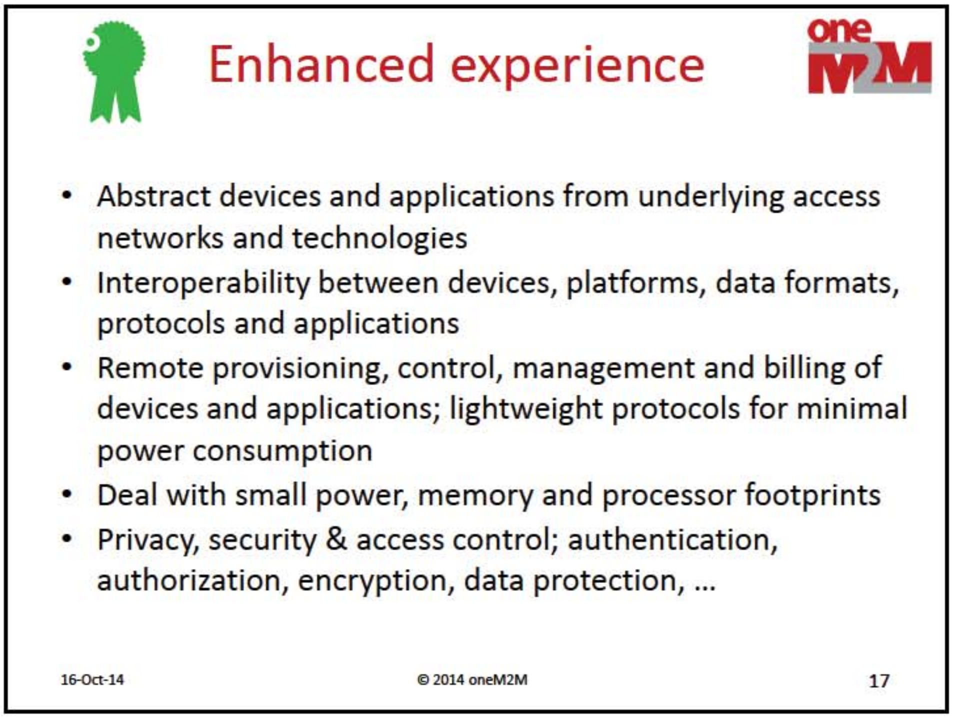
16-Oct-14 © 2014 oneM2M 15



Partnering choices and opportunities

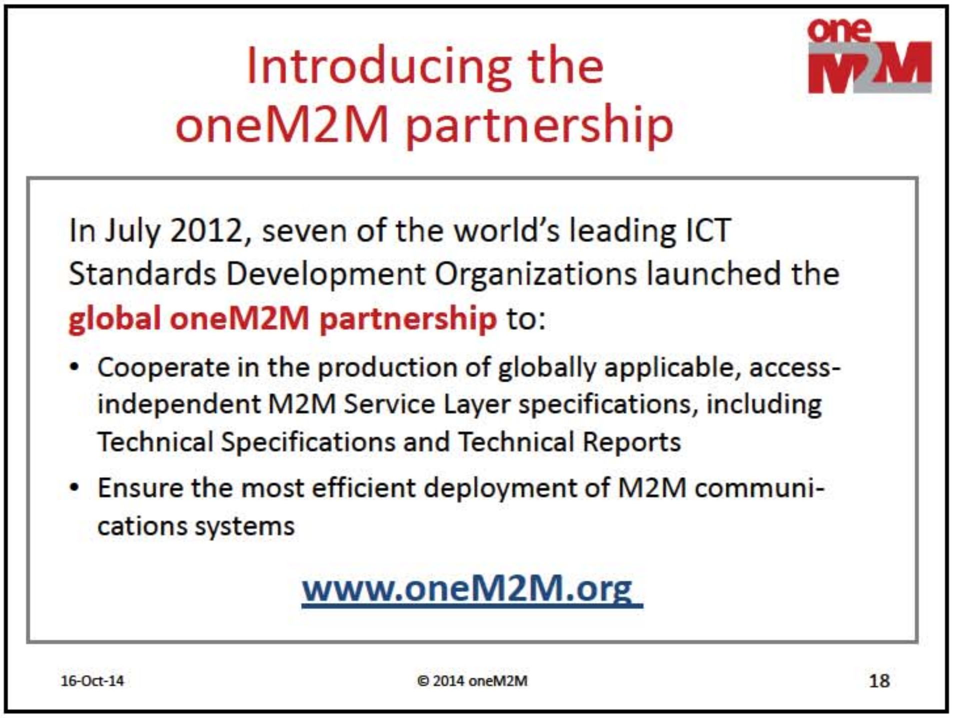
- More suppliers to source M2M solution components from
- More providers who can develop and integrate M2M solutions and applications
- Partnering with other stakeholders to store, discover, access, exchange and share data and content
- Partnering with wireline and wireless service providers and extract more value from underlying networks
- Cross-vendor device configuration and management

16-Oct-14 © 2014 oneM2M 16



Enhanced experience

- Abstract devices and applications from underlying access networks and technologies
- Interoperability between devices, platforms, data formats, protocols and applications
- Remote provisioning, control, management and billing of devices and applications; lightweight protocols for minimal power consumption
- Deal with small power, memory and processor footprints
- Privacy, security & access control; authentication, authorization, encryption, data protection, …

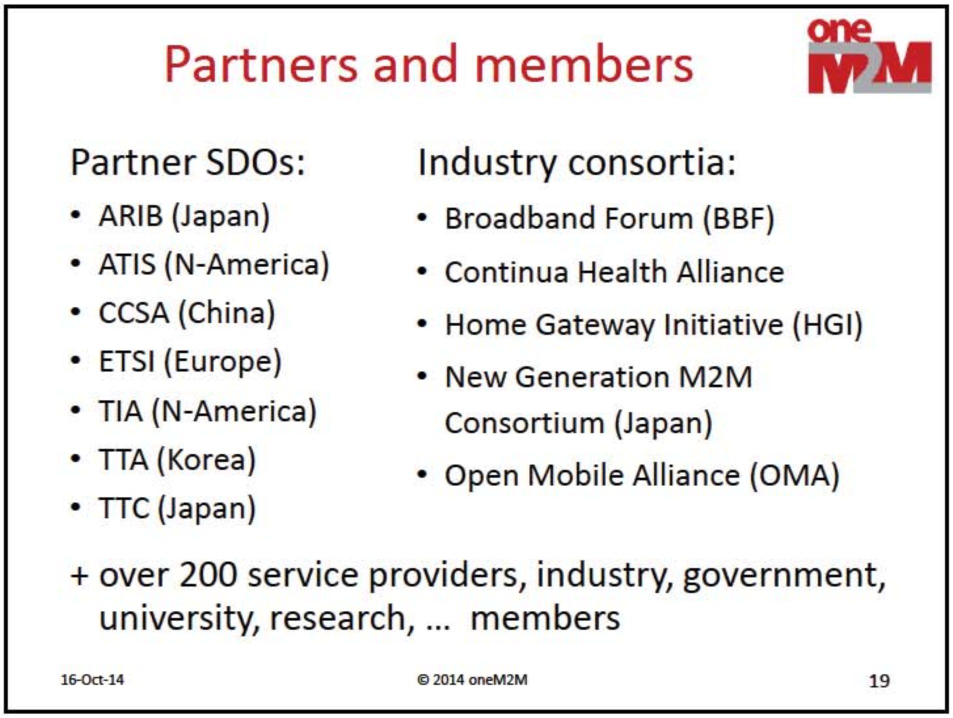


Introducing the oneM2M partnership

In July 2012, seven of the world's leading ICT Standards Development Organizations launched the **global oneM2M partnership** to:

- Cooperate in the production of globally applicable, access-independent M2M Service Layer specifications, including Technical Specifications and Technical Reports
- Ensure the most efficient deployment of M2M communications systems

www.oneM2M.org



Partners and members

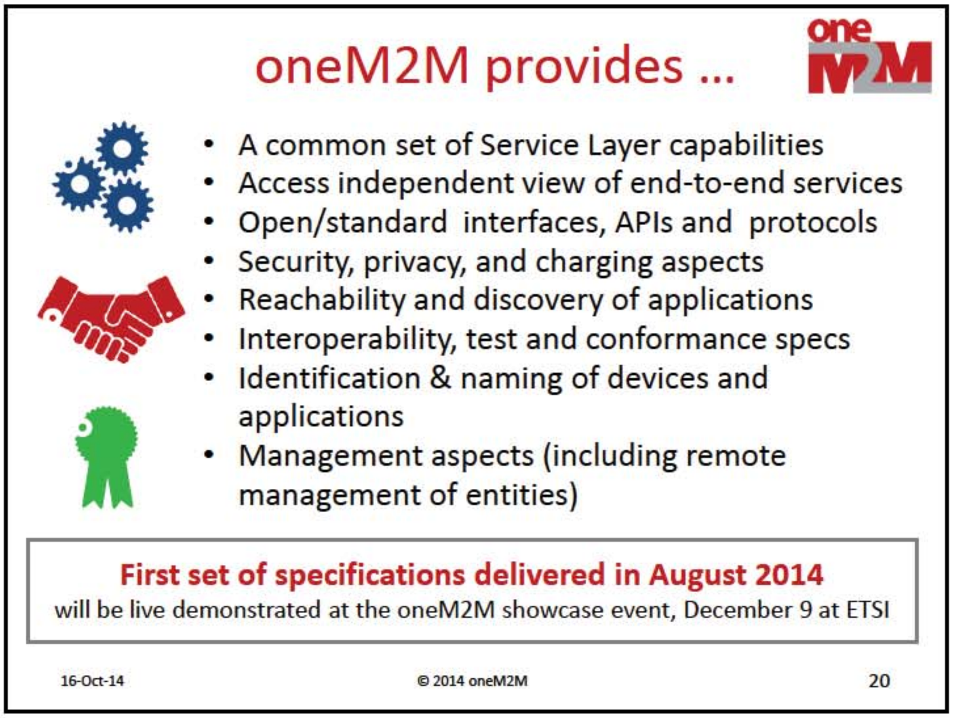
Partner SDOs:

- ARIB (Japan)
- ATIS (N-America)
- CCSA (China)
- ETSI (Europe)
- TIA (N-America)
- TTA (Korea)
- TTC (Japan)

Industry consortia:

- Broadband Forum (BBF)
- Continua Health Alliance
- Home Gateway Initiative (HGI)
- New Generation M2M Consortium (Japan)
- Open Mobile Alliance (OMA)

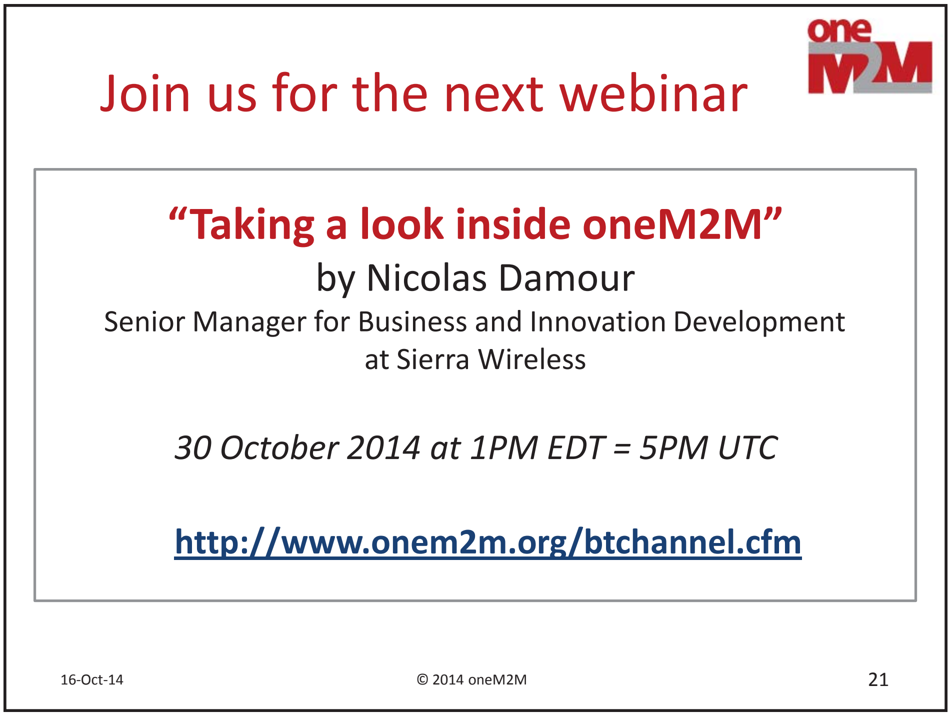
+ over 200 service providers, industry, government, university, research, … members



oneM2M provides …

- A common set of Service Layer capabilities
- Access independent view of end-to-end services
- Open/standard interfaces, APIs and protocols
- Security, privacy, and charging aspects
- Reachability and discovery of applications
- Interoperability, test and conformance specs
- Identification & naming of devices and applications
- Management aspects (including remote management of entities)

First set of specifications delivered in August 2014
will be live demonstrated at the oneM2M showcase event, December 9 at ETSI



Join us for the next webinar

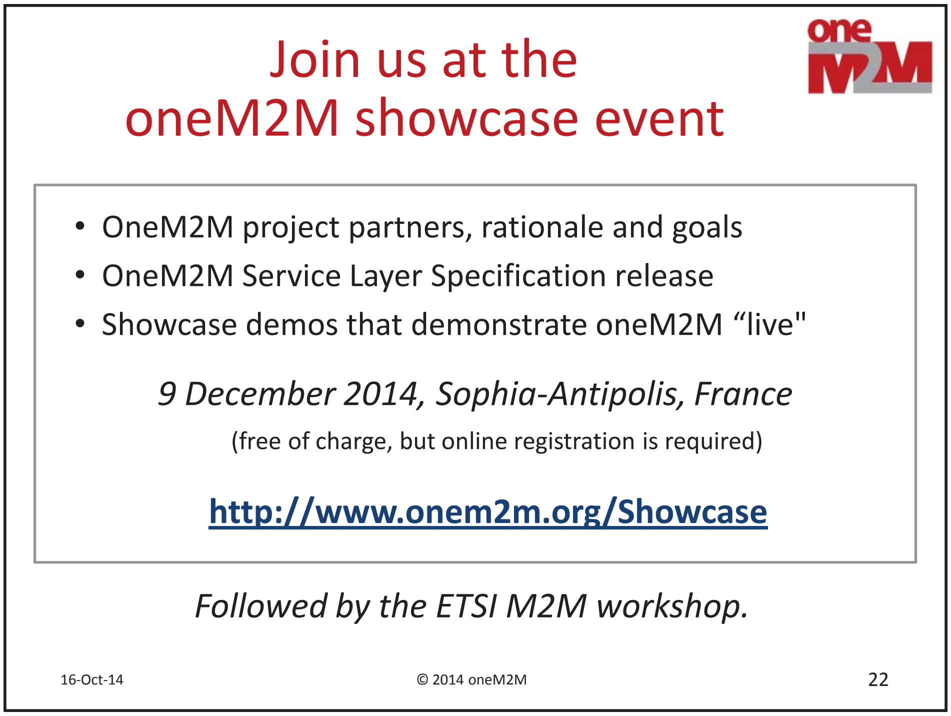
“Taking a look inside oneM2M”

by Nicolas Damour

Senior Manager for Business and Innovation Development
at Sierra Wireless

30 October 2014 at 1PM EDT = 5PM UTC

http://www.onem2m.org/btchannel.cfm



Join us at the oneM2M showcase event

- OneM2M project partners, rationale and goals
- OneM2M Service Layer Specification release
- Showcase demos that demonstrate oneM2M “live"

9 December 2014, Sophia-Antipolis, France

(free of charge, but online registration is required)

http://www.onem2m.org/Showcase

Followed by the ETSI M2M workshop.

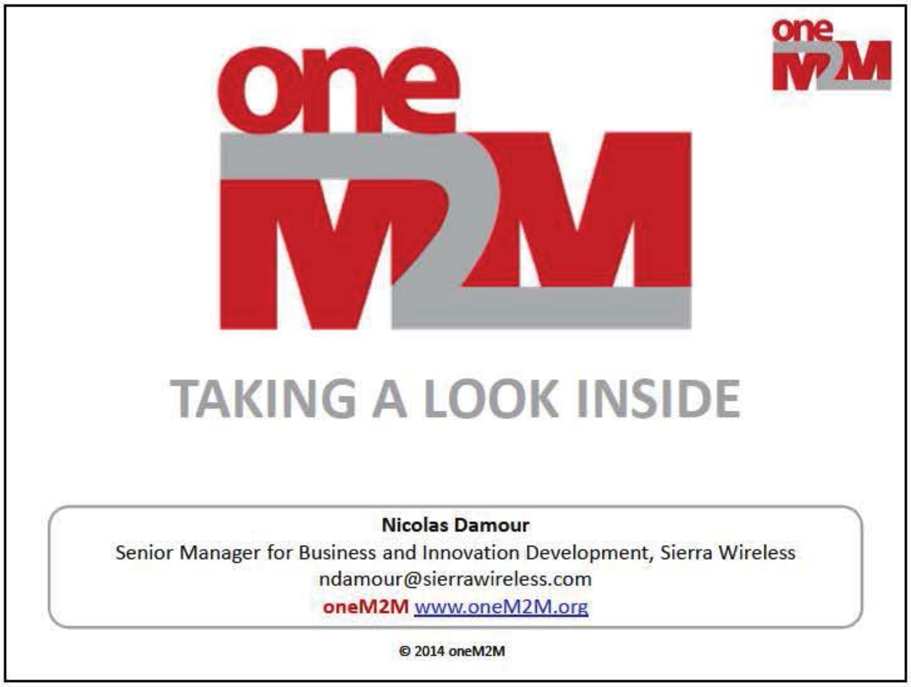
Thank You!

Q&A

16-Oct-14

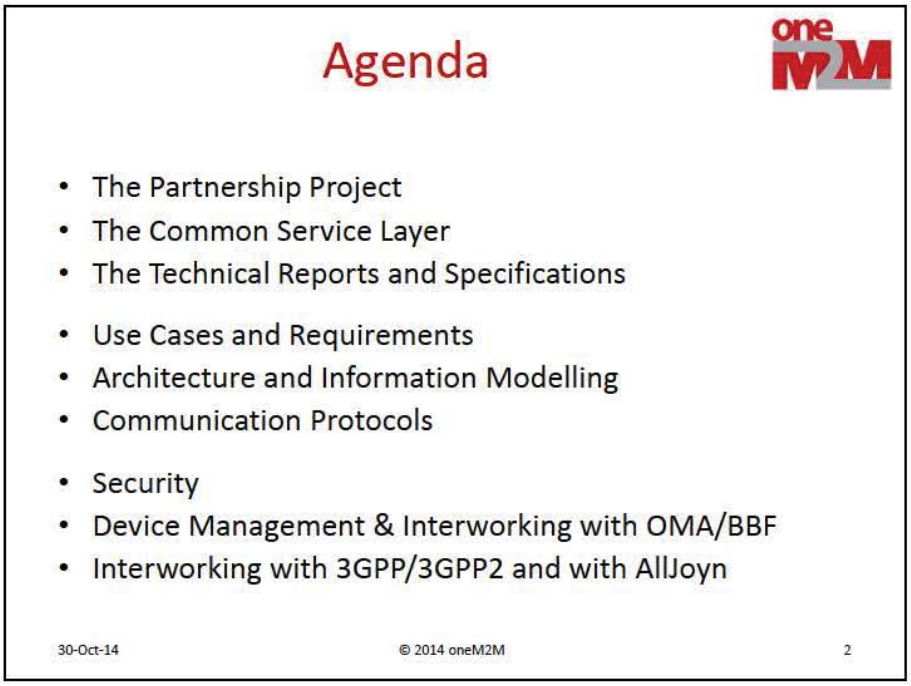
EXHIBIT 1 TO FELDMAN REPLY DECLARATION

DOCUMENT 70



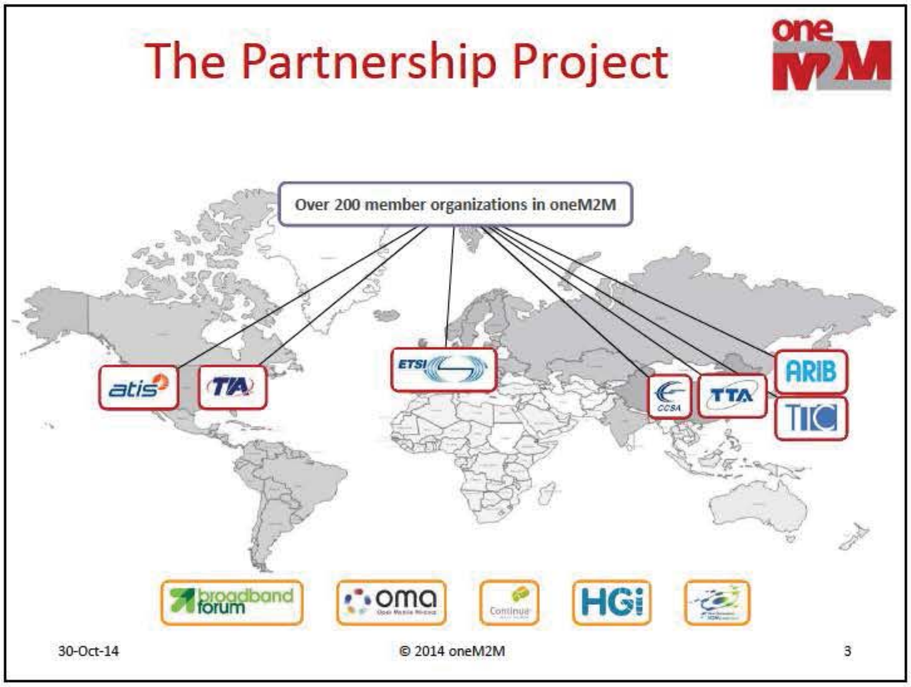
TAKING A LOOK INSIDE

Nicolas Damour
Senior Manager for Business and Innovation Development, Sierra Wireless
ndamour@sierrawireless.com
oneM2M www.oneM2M.org



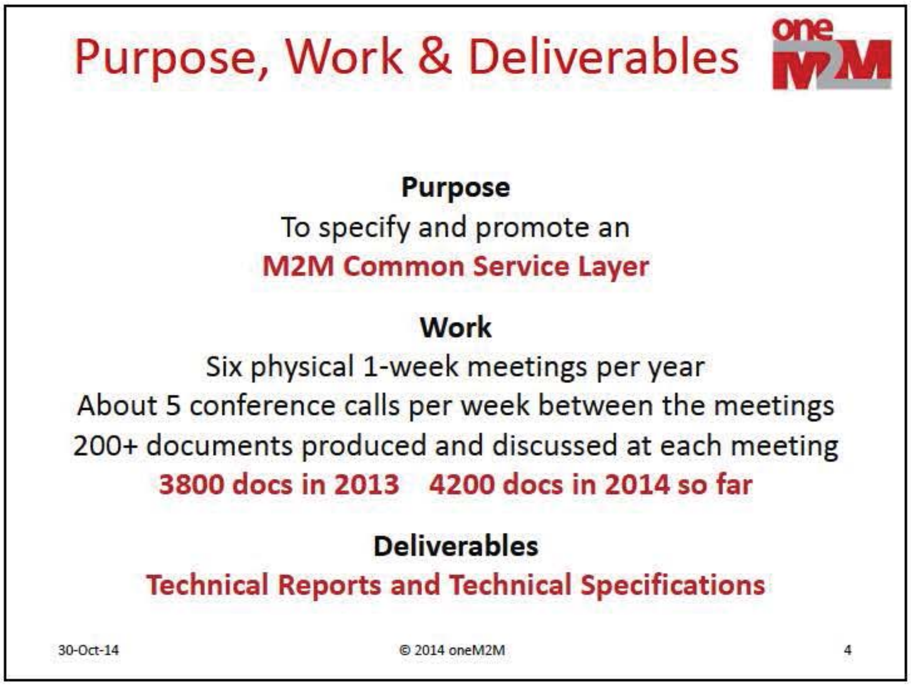
Agenda

- The Partnership Project
- The Common Service Layer
- The Technical Reports and Specifications
- Use Cases and Requirements
- Architecture and Information Modelling
- Communication Protocols
- Security
- Device Management & Interworking with OMA/BBF
- Interworking with 3GPP/3GPP2 and with AllJoyn



The Partnership Project

Over 200 member organizations in oneM2M



Purpose, Work & Deliverables

Purpose

To specify and promote an

M2M Common Service Layer

Work

Six physical 1-week meetings per year

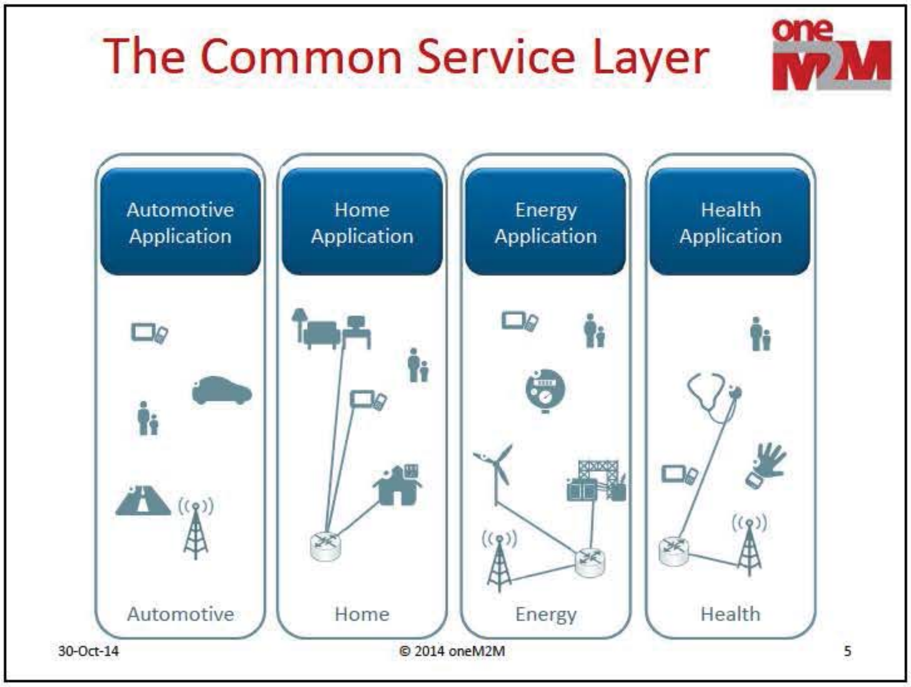
About 5 conference calls per week between the meetings

200+ documents produced and discussed at each meeting

3800 docs in 2013 4200 docs in 2014 so far

Deliverables

Technical Reports and Technical Specifications



The Common Service Layer

Automotive Application

Home Application

Energy Application

Health Application

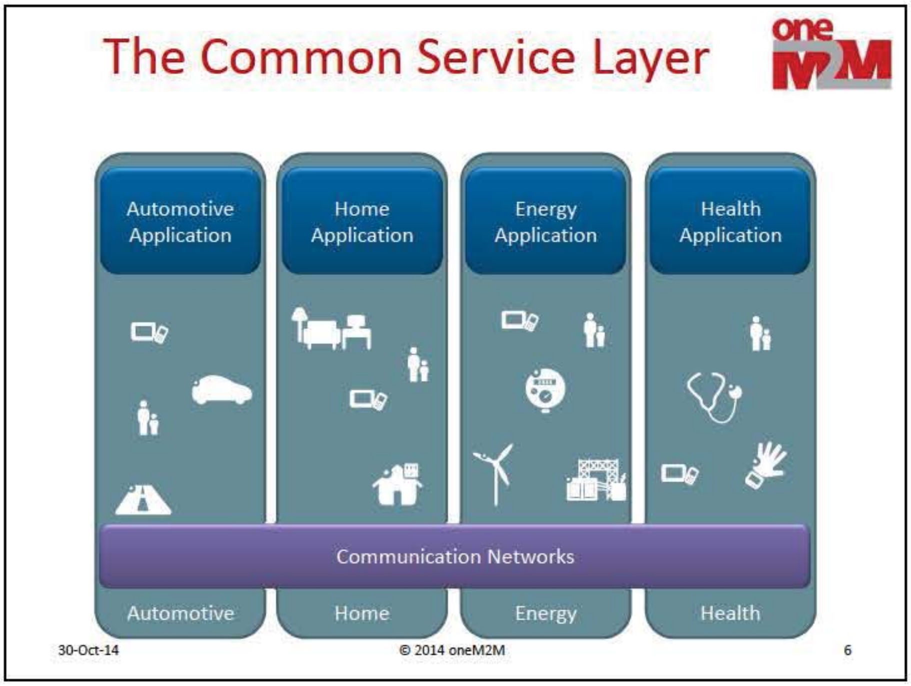
Automotive

Home

Energy

Health

30-Oct-14



The Common Service Layer

Automotive Application

Home Application

Energy Application

Health Application

Communication Networks

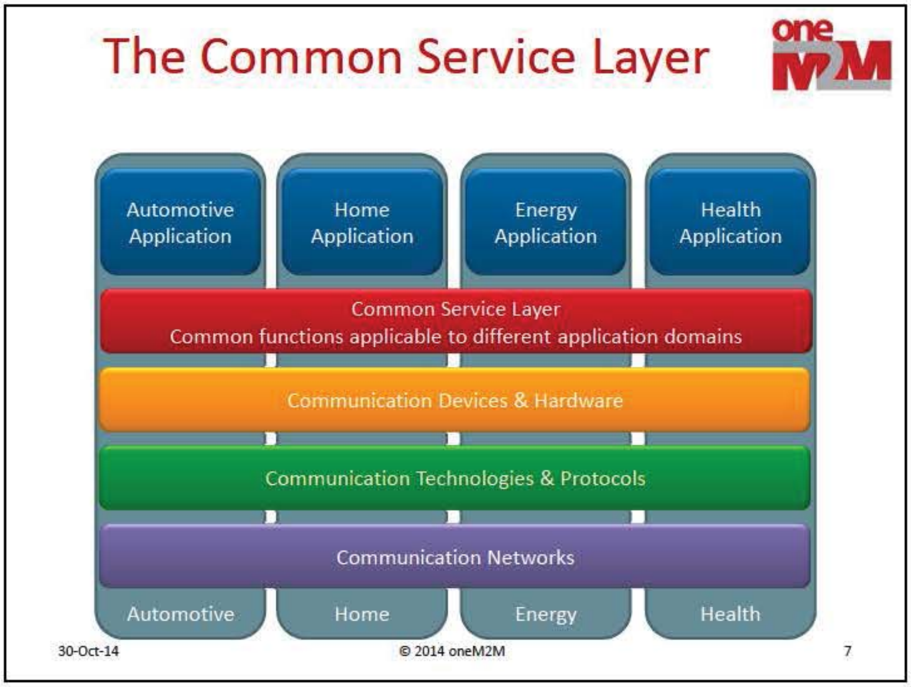
Automotive

Home

Energy

Health

30-Oct-14



The Common Service Layer

Automotive Application

Home Application

Energy Application

Health Application

Common Service Layer
Common functions applicable to different application domains

Communication Devices & Hardware

Communication Technologies & Protocols

Communication Networks

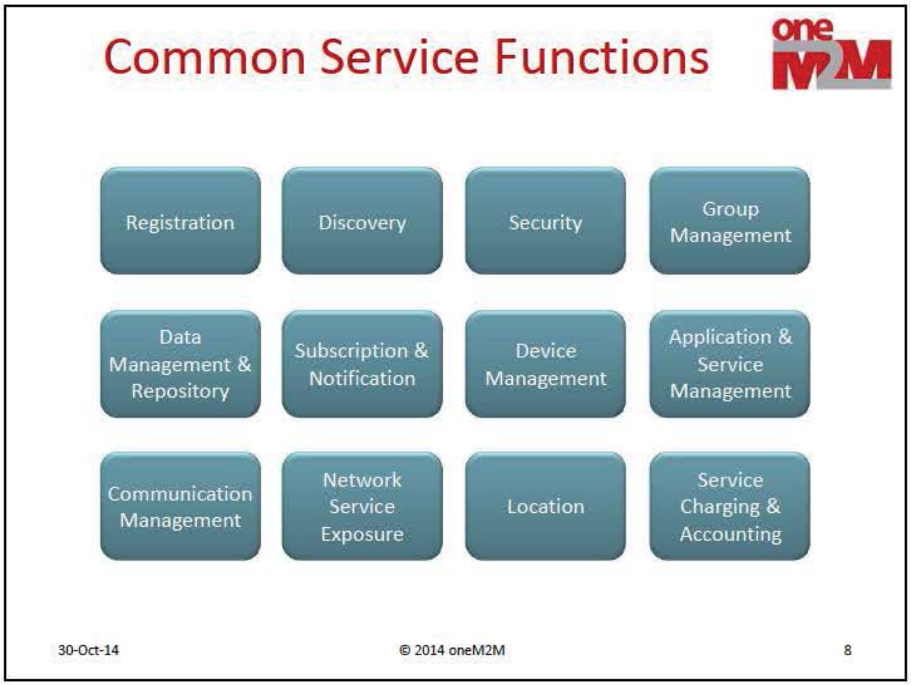
Automotive

Home

Energy

Health

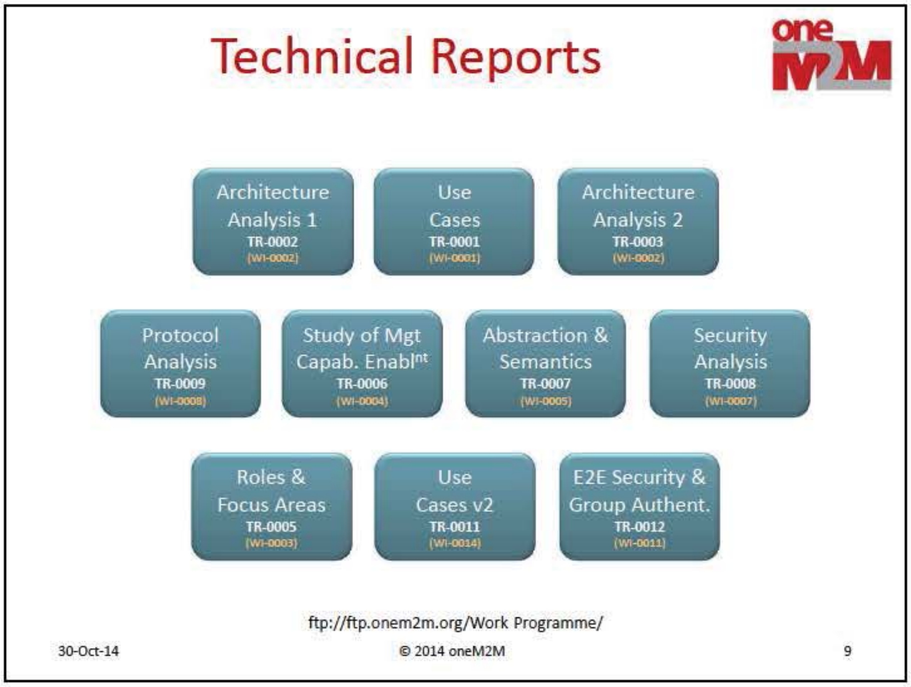
30-Oct-14 © 2014 oneM2M 7



Common Service Functions

Registration	Discovery	Security	Group Management
Data Management & Repository	Subscription & Notification	Device Management	Application & Service Management
Communication Management	Network Service Exposure	Location	Service Charging & Accounting

30-Oct-14 © 2014 oneM2M 8



Technical Reports

Architecture Analysis 1
TR-0002
(WI-0002)

Use Cases
TR-0001
(WI-0001)

Architecture Analysis 2
TR-0003
(WI-0002)

Protocol Analysis
TR-0009
(WI-0008)

Study of Mgt Capab. Enabl^{nt}
TR-0006
(WI-0004)

Abstraction & Semantics
TR-0007
(WI-0005)

Security Analysis
TR-0008
(WI-0007)

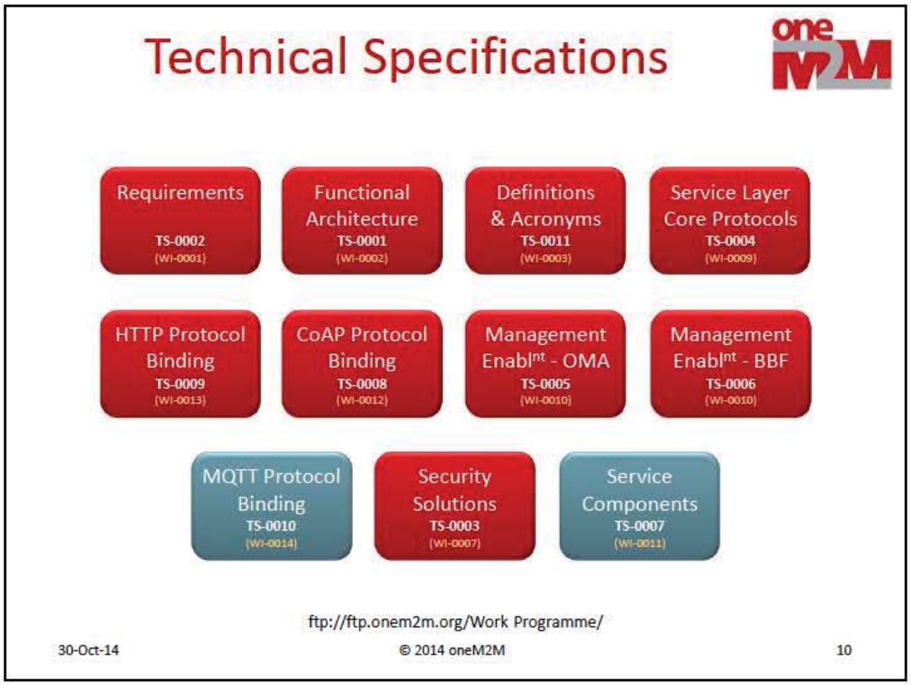
Roles & Focus Areas
TR-0005
(WI-0003)

Use Cases v2
TR-0011
(WI-0014)

E2E Security & Group Authent.
TR-0012
(WI-0011)

ftp://ftp.onem2m.org/Work Programme/

30-Oct-14 © 2014 oneM2M 9



Technical Specifications

Requirements
TS-0002
(WI-0001)

Functional Architecture
TS-0001
(WI-0002)

Definitions & Acronyms
TS-0011
(WI-0003)

Service Layer Core Protocols
TS-0004
(WI-0009)

HTTP Protocol Binding
TS-0009
(WI-0013)

CoAP Protocol Binding
TS-0008
(WI-0012)

Management Enabl^{nt} - OMA
TS-0005
(WI-0010)

Management Enabl^{nt} - BBF
TS-0006
(WI-0010)

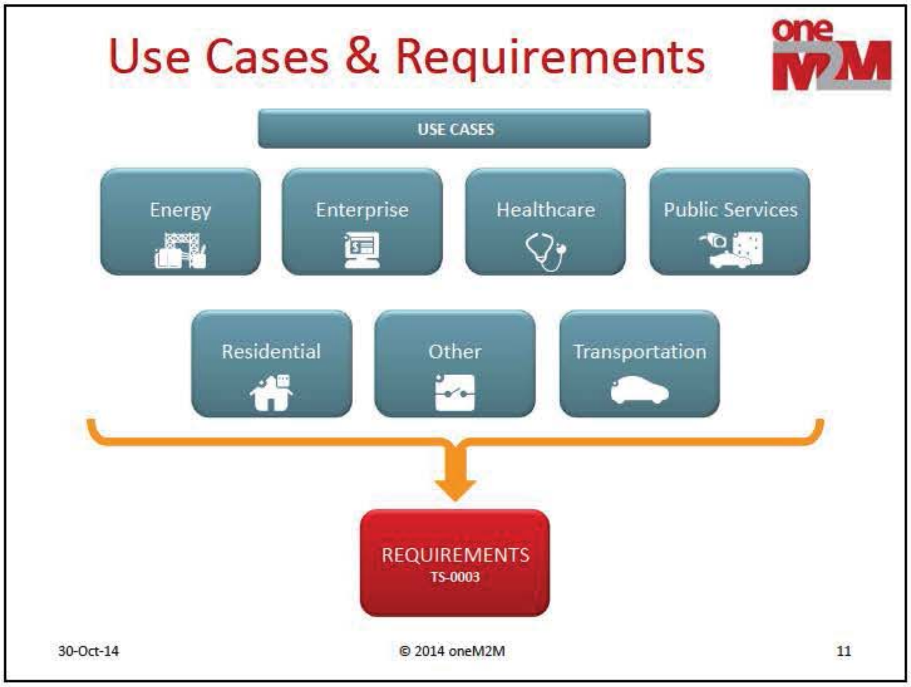
MQTT Protocol Binding
TS-0010
(WI-0014)

Security Solutions
TS-0003
(WI-0007)

Service Components
TS-0007
(WI-0011)

ftp://ftp.onem2m.org/Work Programme/

30-Oct-14 © 2014 oneM2M 10



Use Cases & Requirements

USE CASES

Energy

Enterprise

Healthcare

Public Services

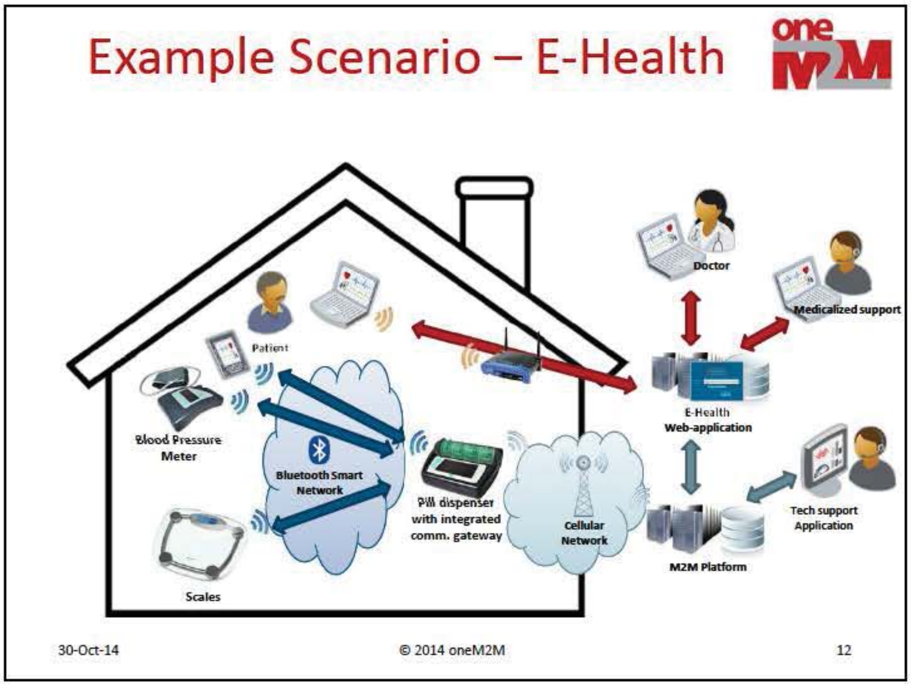
Residential

Other

Transportation

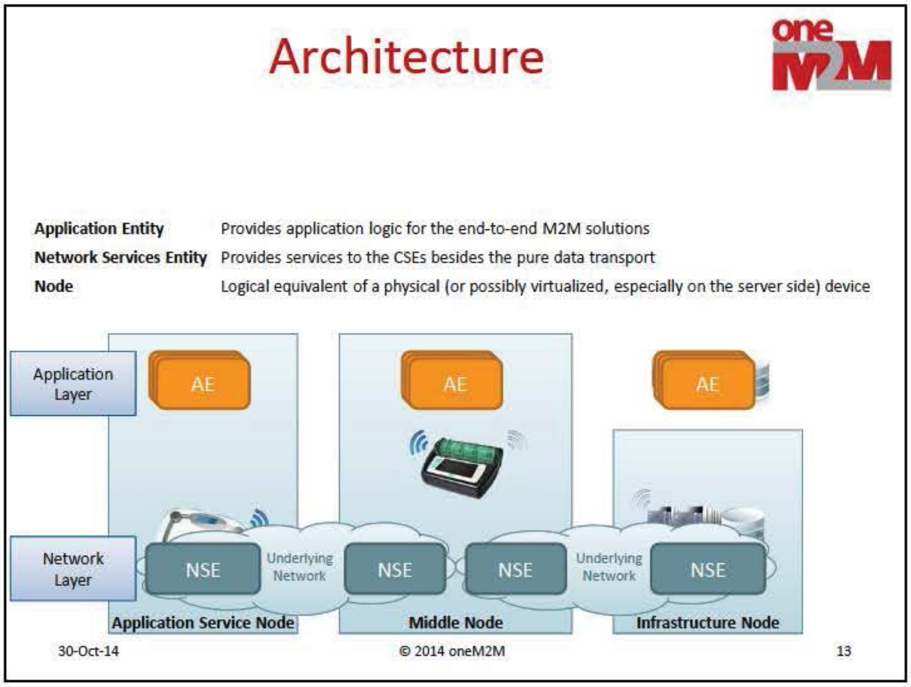
REQUIREMENTS
TS-0003

30-Oct-14 © 2014 oneM2M 11



Example Scenario – E-Health

30-Oct-14 © 2014 oneM2M 12



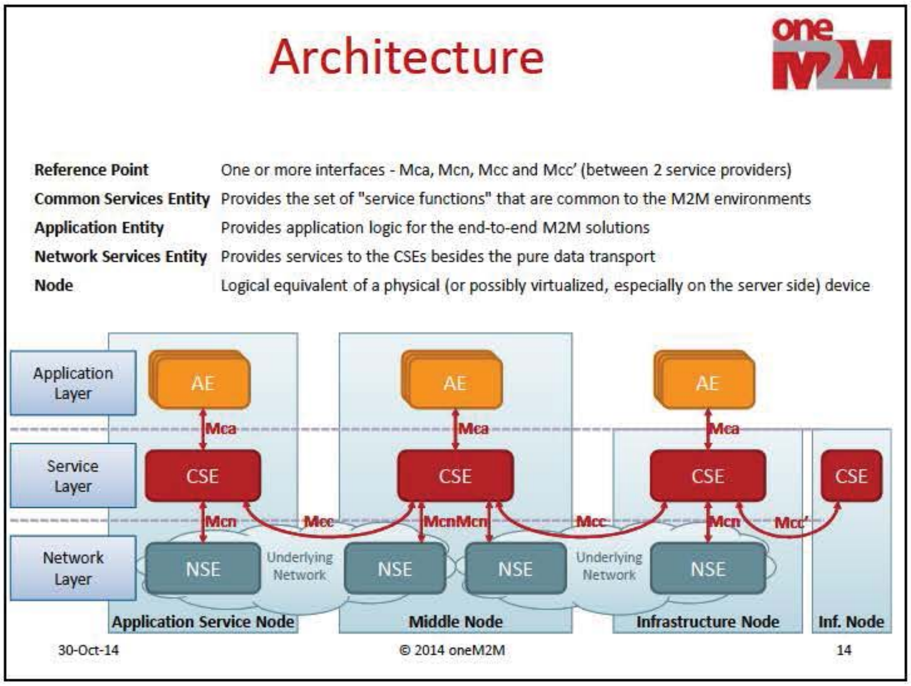
Architecture

Application Entity Provides application logic for the end-to-end M2M solutions

Network Services Entity Provides services to the CSEs besides the pure data transport

Node Logical equivalent of a physical (or possibly virtualized, especially on the server side) device

30-Oct-14 © 2014 oneM2M 13



Architecture

Reference Point One or more interfaces - Mca, Mcn, Mcc and Mcc' (between 2 service providers)

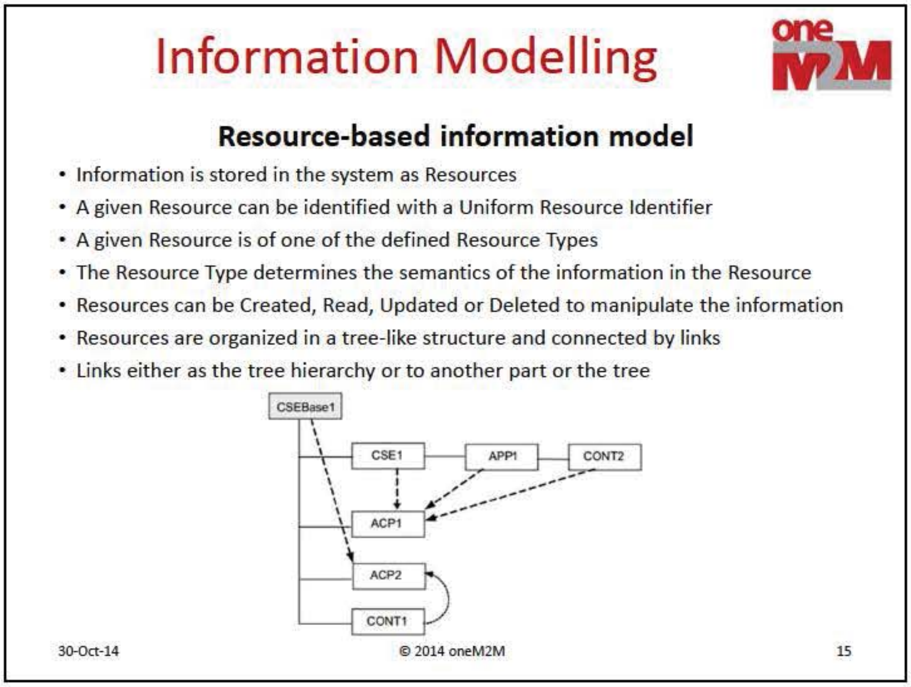
Common Services Entity Provides the set of "service functions" that are common to the M2M environments

Application Entity Provides application logic for the end-to-end M2M solutions

Network Services Entity Provides services to the CSEs besides the pure data transport

Node Logical equivalent of a physical (or possibly virtualized, especially on the server side) device

30-Oct-14 © 2014 oneM2M 14

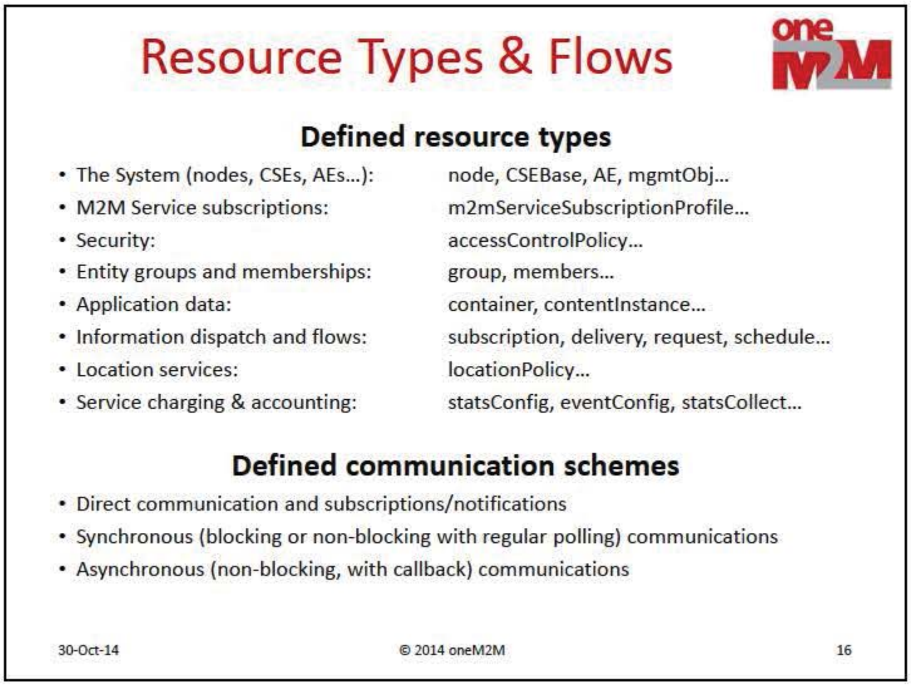


Information Modelling

Resource-based information model

- Information is stored in the system as Resources
- A given Resource can be identified with a Uniform Resource Identifier
- A given Resource is of one of the defined Resource Types
- The Resource Type determines the semantics of the information in the Resource
- Resources can be Created, Read, Updated or Deleted to manipulate the information
- Resources are organized in a tree-like structure and connected by links
- Links either as the tree hierarchy or to another part or the tree

30-Oct-14 © 2014 oneM2M 15



Resource Types & Flows

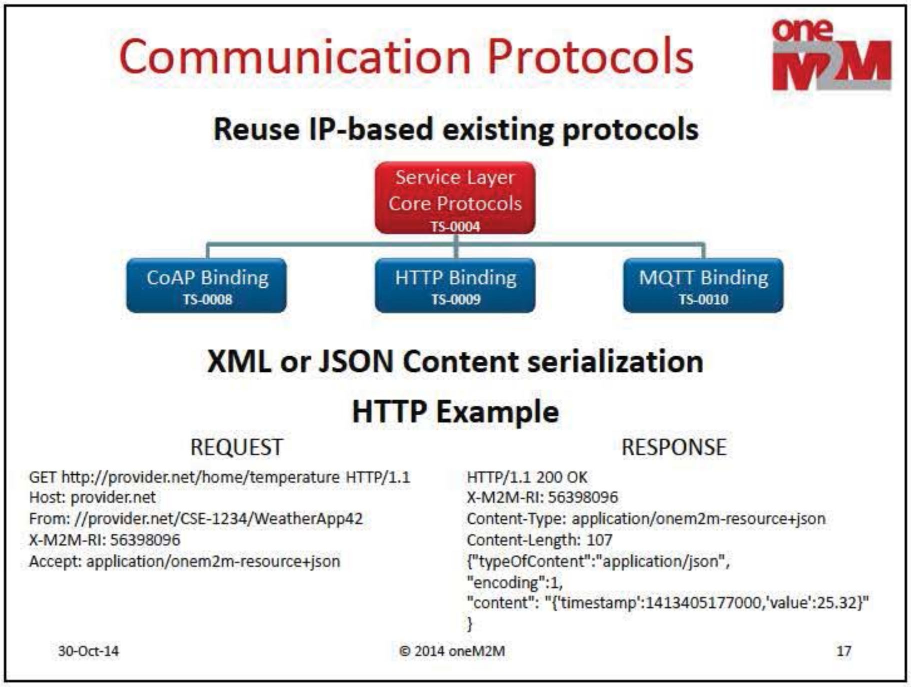
Defined resource types

• The System (nodes, CSEs, AEs…):	node, CSEBase, AE, mgmtObj…
• M2M Service subscriptions:	m2mServiceSubscriptionProfile…
• Security:	accessControlPolicy…
• Entity groups and memberships:	group, members…
• Application data:	container, contentInstance…
• Information dispatch and flows:	subscription, delivery, request, schedule…
• Location services:	locationPolicy…
• Service charging & accounting:	statsConfig, eventConfig, statsCollect…

Defined communication schemes

- Direct communication and subscriptions/notifications
- Synchronous (blocking or non-blocking with regular polling) communications
- Asynchronous (non-blocking, with callback) communications

30-Oct-14 © 2014 oneM2M 16



Communication Protocols

Reuse IP-based existing protocols

Service Layer Core Protocols
TS-0004

CoAP Binding
TS-0008

HTTP Binding
TS-0009

MQTT Binding
TS-0010

XML or JSON Content serialization

HTTP Example

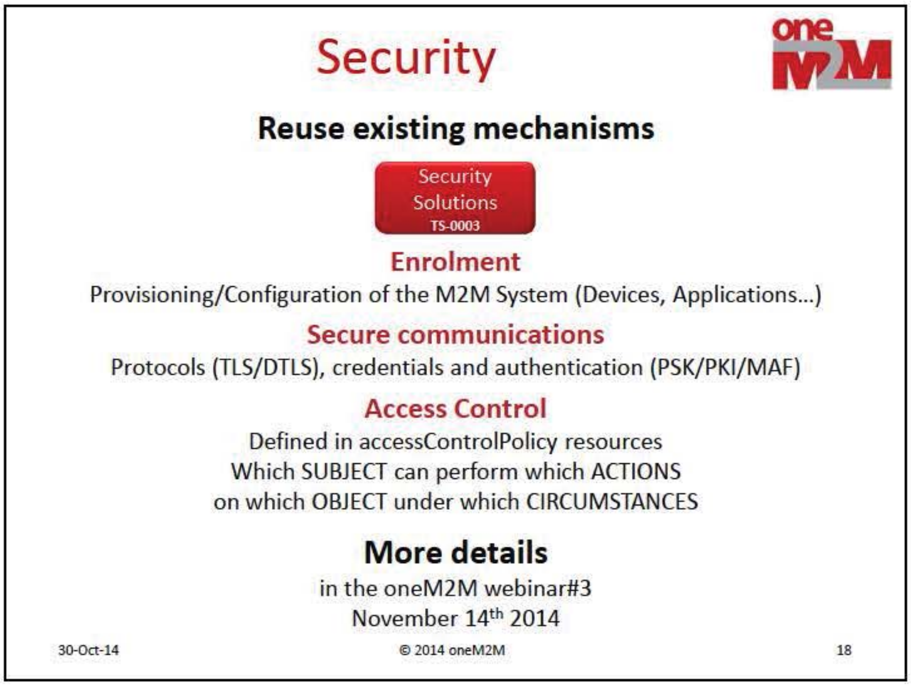
REQUEST

```
GET http://provider.net/home/temperature HTTP/1.1
Host: provider.net
From: //provider.net/CSE-1234/WeatherApp42
X-M2M-RI: 56398096
Accept: application/onem2m-resource+json
```

RESPONSE

```
HTTP/1.1 200 OK
X-M2M-RI: 56398096
Content-Type: application/onem2m-resource+json
Content-Length: 107
{"typeOfContent":"application/json",
"encoding":1,
"content": "{'timestamp':1413405177000,'value':25.32}"
}
```

30-Oct-14 © 2014 oneM2M 17



Security

Reuse existing mechanisms

Security Solutions
TS-0003

Enrolment

Provisioning/Configuration of the M2M System (Devices, Applications…)

Secure communications

Protocols (TLS/DTLS), credentials and authentication (PSK/PKI/MAF)

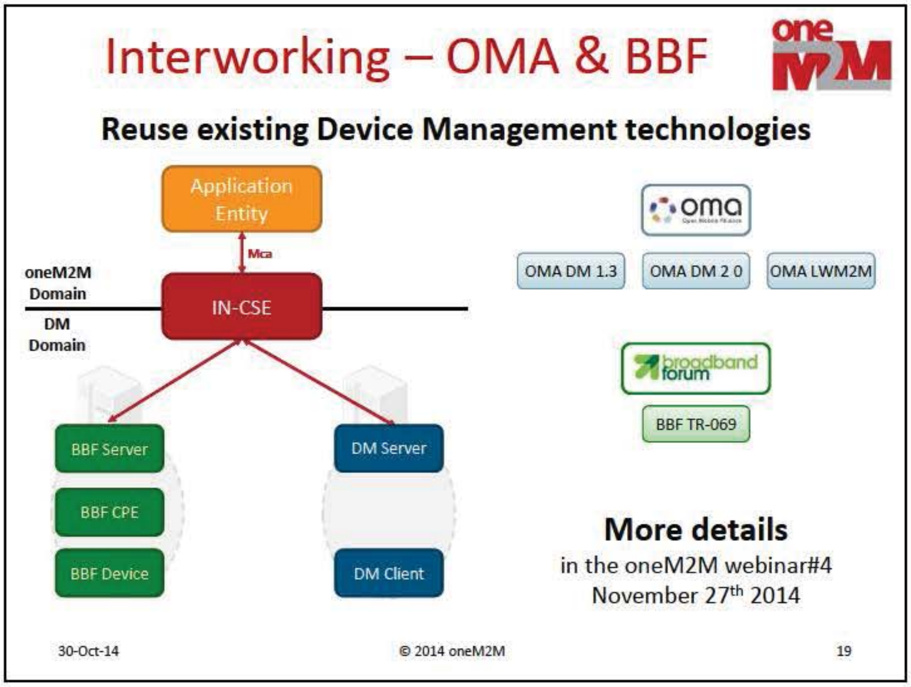
Access Control

Defined in accessControlPolicy resources
Which SUBJECT can perform which ACTIONS
on which OBJECT under which CIRCUMSTANCES

More details

in the oneM2M webinar#3
November 14th 2014

30-Oct-14 © 2014 oneM2M 18



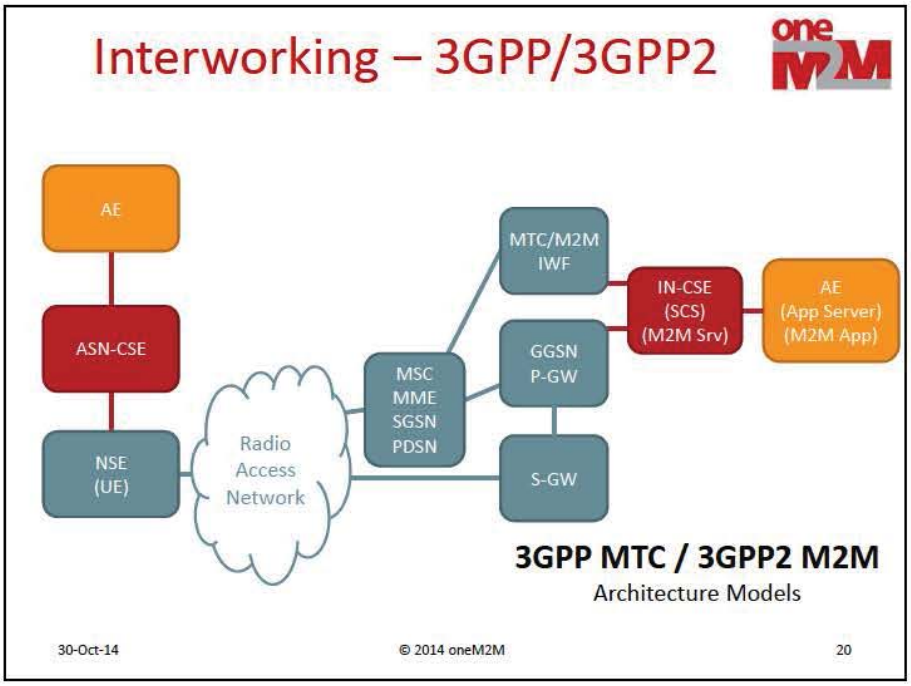
Interworking – OMA & BBF

Reuse existing Device Management technologies

More details

in the oneM2M webinar#4
November 27th 2014

30-Oct-14 © 2014 oneM2M 19

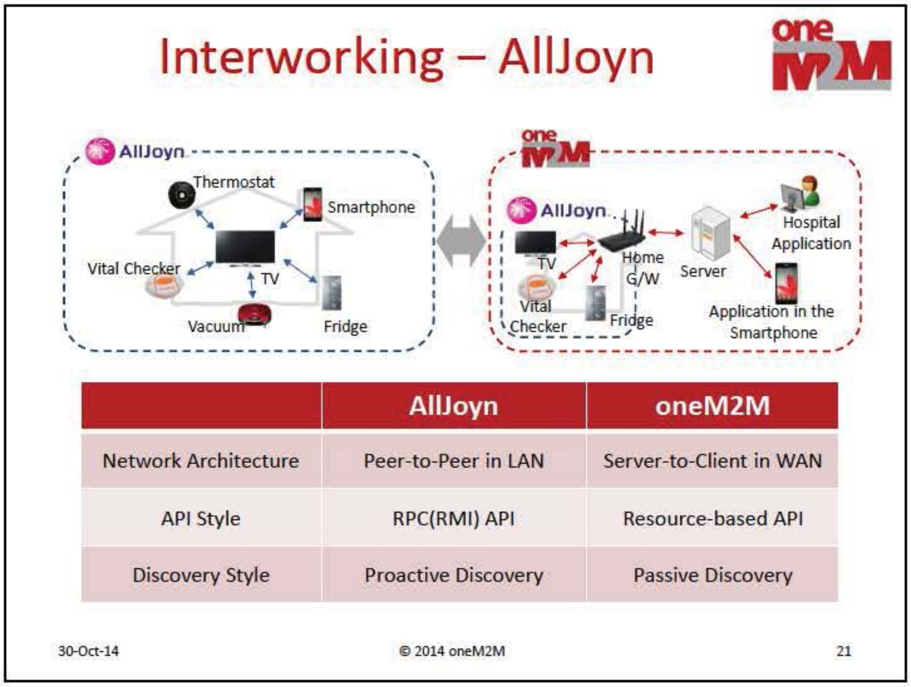


Interworking – 3GPP/3GPP2

3GPP MTC / 3GPP2 M2M

Architecture Models

30-Oct-14 © 2014 oneM2M 20

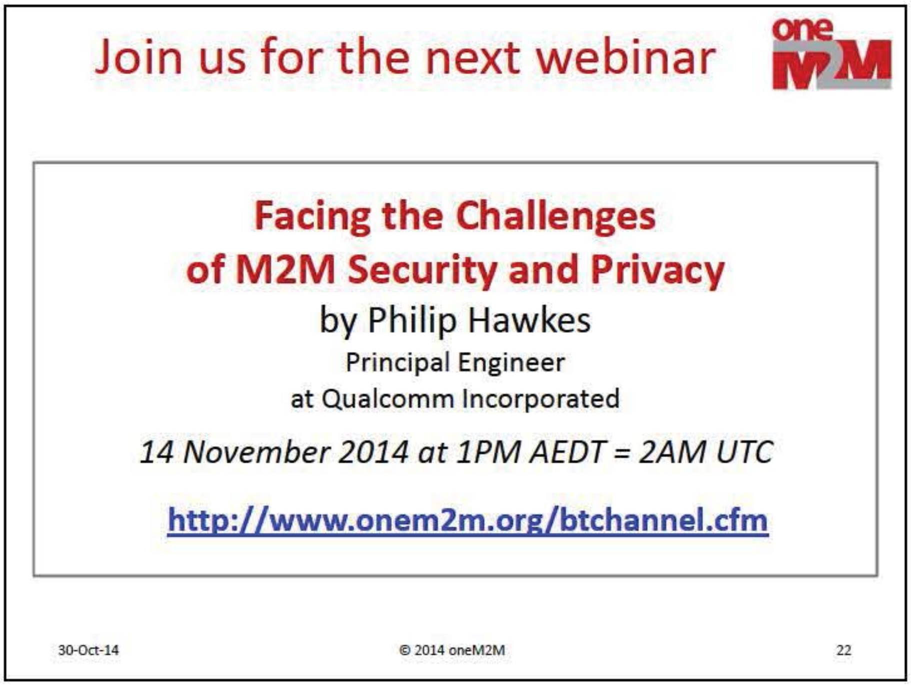


Interworking – AllJoyn

AllJoyn
Thermostat
Smartphone
Vital Checker
TV
Vacuum
Fridge
oneM2M
AllJoyn
TV
Home G/W
Server
Hospital Application
Vital Checker
Fridge
Application in the Smartphone

	AllJoyn	oneM2M
Network Architecture	Peer-to-Peer in LAN	Server-to-Client in WAN
API Style	RPC(RMI) API	Resource-based API
Discovery Style	Proactive Discovery	Passive Discovery

30-Oct-14 21



Join us for the next webinar

Facing the Challenges of M2M Security and Privacy

by Philip Hawkes

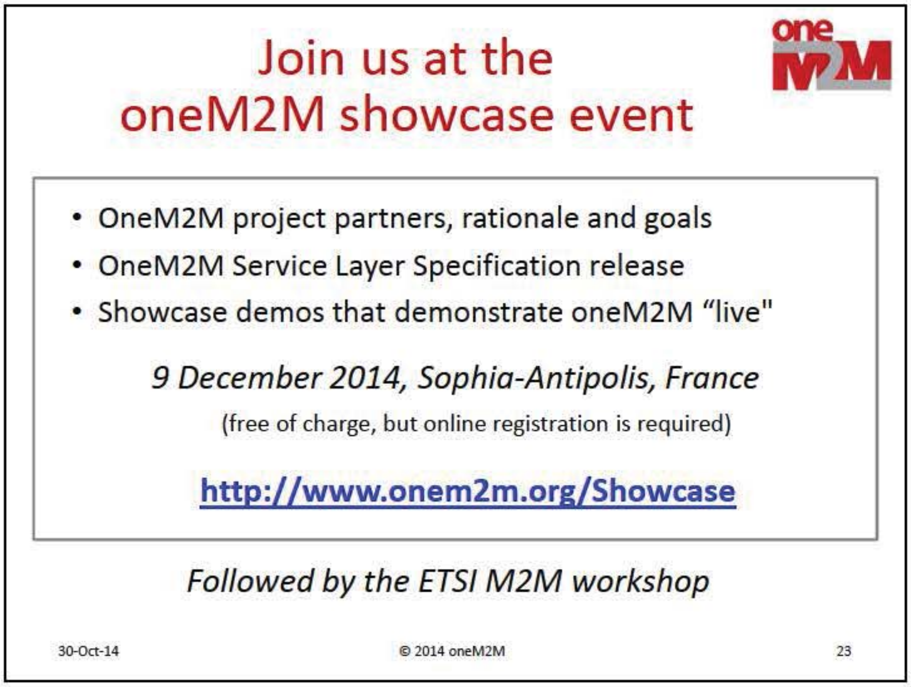
Principal Engineer

at Qualcomm Incorporated

14 November 2014 at 1PM AEDT = 2AM UTC

http://www.onem2m.org/btchannel.cfm

30-Oct-14 22



Join us at the oneM2M showcase event

- OneM2M project partners, rationale and goals
- OneM2M Service Layer Specification release
- Showcase demos that demonstrate oneM2M “live"

9 December 2014, Sophia-Antipolis, France

(free of charge, but online registration is required)

http://www.onem2m.org/Showcase

Followed by the ETSI M2M workshop

Thank You!

Q&A

EXHIBIT 1 TO FELDMAN REPLY DECLARATION

DOCUMENT 71

ETSI TS 133 220 V12.3.0 (2014-10)

TECHNICAL SPECIFICATION

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Generic Authentication Architecture (GAA);
Generic Bootstrapping Architecture (GBA)
(3GPP TS 33.220 version 12.3.0 Release 12)

Reference

RTS/TSGS-0333220vc30

Keywords

GSM, LTE, SECURITY, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes the security features and mechanisms to bootstrap authentication and key agreement for application security. Candidate applications to use this bootstrapping mechanism include but are not restricted to subscriber certificate distribution TS 33.221 [5]. Subscriber certificates support services whose provision mobile operator assists, as well as services that mobile operator provides.

The scope of this specification includes generic bootstrapping functions, an architecture overview and the detailed procedure how to bootstrap the credential.

Clause 4 of this specification describes a mechanism, called GBA_ME, to bootstrap authentication and key agreement, which does not require any changes to the UICC. Clause 5 of this specification describes a mechanism, called GBA_U, to bootstrap authentication and key agreement, which does require changes to the UICC, but provides enhanced security by storing certain derived keys on the UICC. Annex I of this specification describes a mechanism, called 2G GBA, to bootstrap authentication and key agreement using 2G AKA protocol. Annex M of this specification describes a mechanism, called GBA_Digest, to bootstrap authentication and key agreement using HTTP Digest protocol with SIP Digest credentials.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 31.102: "Characteristics of the USIM application".

[2] 3GPP TS 33.102: "3G Security; Security architecture".

[3] IETF RFC 2617: "HTTP Authentication: Basic and Digest Access Authentication".

[4] IETF RFC 3310: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[5] 3GPP TS 33.221: "Generic Authentication Architecture (GAA); Support for Subscriber Certificates".

[6] Void

[7] Void

[8] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2 (Release 6)".

[9] Void.

[10] 3GPP TS 31.103: "Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "Numbering, addressing and identification".

[12] IETF RFC 3548: "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3G Security; Network domain security; IP network layer security".

[14] Void.

[15] 3GPP TS 31.101: "UICC-terminal interface; Physical and logical characteristics".

[16] 3GPP TS 33.203: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services".

[17] Void.

[18] IETF RFC 2818: "HTTP over TLS".

[19] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[20] Void.

[21] Void.

[22] IETF RFC 2104: "HMAC: Keyed-Hashing for Message Authentication".

[23] ISO/IEC 10118-3:2004: "Information Technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions".

[24] IETF RFC 3629: "UTF-8, a transformation format of ISO 10646".

[25] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

[26] 3GPP TS 33.246: "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)".

[27] Void.

[28] IETF RFC 2246: "The TLS Protocol Version 1".

[29] 3GPP TS 24.109: "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details".

[30] (void)

[31] (void)

[32] 3GPP TS 29.109: "Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on the Diameter protocol; Stage 3".

[33] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

[34] 3GPP TS 23.002: 'Network architecture '.

[35] 3GPP TS 33.401: "3GPP System Architecture Evolution (SAE); Security Architecture".

[36] 3GPP TS 33.402: "3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses".

[37] "Unicode Standard Annex #15; Unicode Normalization Forms", Unicode 5.1.0, March 2008. http://www.unicode.org

[38] 3GPP TS 26.237: "IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast Service (MBMS) User Service; Protocols".

[39] 3GPP TS 33.224: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push Layer".

[40] 3GPP TS 33.328: "IMS Media plane security".

[41] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[42] (void)

[43] Void.

[44] IETF RFC 5705: "Keying Material Exporters for Transport Layer Security (TLS)".

[45] 3GPP TS 33.223: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push function".

[46] 3GPP TS 44.006 "Technical Specification Group GSM/EDGE Radio Access Network; Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification"

[47] 3GPP TS 43.020 "Technical Specification Group Services and system Aspects; Security related network functions"

[48] IETF RFC 5929 "Channel Bindings for TLS"

[49] 3GPP TS 33.303: "Proximity-based Services; Security Aspects"

3 Definitions, abbreviations symbols and conventions

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

AKA-based GBA: This term collectively refers to all GBA variants that make use of a form of the AKA protocol on the Ub interface, i.e. the term refers to GBA_ME, GBA_U, and 2G GBA, as defined in the present document, and to GBA_push as defined in TS 33.223 [45].

Application: In all places in this document where the term application is used to refer to a service offered by the MNO or a third party to the mobile subscriber, then it always denotes the type of application and not the actual instance of an application installed on an application server.

Bootstrapping Server Function: BSF is hosted in a network element under the control of an MNO. BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by running the bootstrapping procedure. The shared secret can be used between NAFs and UEs, for example, for authentication purposes.

Bootstrapping Usage Procedure: A procedure using bootstrapped security association over Ua reference point.

GBA Function: A function on the ME executing the bootstrapping procedure with BSF (i.e. supporting the Ub reference point) and providing Ua applications with security association to run bootstrapping usage procedure. GBA function is called by a Ua application when a Ua application wants to use bootstrapped security association.

ME-based GBA: in GBA_ME, all GBA-specific functions are carried out in the ME. The UICC is GBA-unaware. If the term GBA is used in this document without any further qualification then always GBA_ME is meant, see clause 4 of this specification.

UICC-based GBA: this is a GBA with UICC-based enhancement. In GBA_U, the GBA-specific functions are split between ME and UICC, see clause 5 of this specification.

GBA_Digest: A GBA variant that extends the usage of GBA to environments where the UICC is not available to the subscriber. In this variant, the GBA client on the UE and the BSF communicate using HTTP protocol and SIP Digest credentials, such as a shared secret or password, that are used for authentication instead of credentials stored in the SIM, USIM or ISIM.

Network Application Function: NAF is hosted in a network element. GBA may be used between NAFs and UEs for authentication purposes, and for securing the communication path between the UE and the NAF.

Bootstrapping Transaction Identifier: the bootstrapping transaction identifier (B-TID) is used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

GBA User Security Settings: GUSS contains the BSF specific information element and the set of all application-specific USSs.

GUSS timestamp: the timestamp of the GUSS is set by the HSS. It changes whenever the HSS has modified the GUSS.

NAF Group: A grouping of NAFs to allow assignment of different USSs to NAFs representing the same application. This grouping is done in each home network separately, i.e. one NAF contacting BSFs in different home networks belongs to different groups in every home network.

NAF_Id: The FQDN of the NAF, concatenated with the Ua security protocol identifier.

Temporary IP Multimedia Private Identity: a temporary identity which is used on the Ub interface to prevent passive eavesdropping attacks against the IMPI.

Ua Application: An application on the ME intended to run bootstrapping usage procedure with a NAF.

Ua security protocol identifier: An identifier which is associated with a security protocol over Ua.

User Security Setting: A USS is an application and subscriber specific parameter set that defines two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g. IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). In addition, a USS may contain a key selection indication, which is used in the GBA_U case to mandate the usage of either the ME-based key (Ks_(ext)_NAF) or the UICC-based key (Ks_int_NAF) or both. Sometimes also called application-specific user security setting. The USS is delivered to the BSF as a part of GUSS from the HSS, and from the BSF to the NAF if requested by the NAF.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AK	Anonymity Key
AKA	Authentication and Key Agreement
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
CA	Certificate Authority
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
GUSS	GBA User Security Settings
HLR	Home Location Register
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
Ks_int_NAF	Derived key in GBA_U which remains on UICC
Ks_ext_NAF	Derived key in GBA_U
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure
SLF	Subscriber Locator Function
TMPI	Temporary IP Multimedia Private Identity
USS	User Security Setting

3.3 Symbols

For the purposes of the present document, the following symbols apply:

||	Concatenation
⊕	Exclusive or

3.4 Conventions

All data variables in this specification are presented with the most significant substring on the left hand side and the least significant substring on the right hand side. A substring may be a bit, byte or other arbitrary length bitstring.

Where a variable is broken down into a number of substrings, the leftmost (most significant) substring is numbered 0, the next most significant is numbered 1, and so on through to the least significant.

4 Generic Bootstrapping Architecture

The 3GPP authentication infrastructure, including the 3GPP Authentication Centre (AuC), the USIM or the ISIM, and the 3GPP AKA protocol run between them, is a very valuable asset of 3GPP operators. It has been recognised that this infrastructure could be leveraged to enable application functions in the network and on the user side to establish shared keys. Therefore, 3GPP can provide the "bootstrapping of application security" to authenticate the subscriber by defining a Generic Bootstrapping Architecture (GBA) based on AKA protocol.

4.1 Reference model

For HLR and HSS definitions used in this chapter refer to [34].

When HSS is mentioned in this specification without an indication of supported reference point towards the BSF, then the support of the Zh reference point is meant.

Figure 4.1 shows a simple network model of the entities involved in the bootstrapping approach when an HSS with Zh reference point is deployed, and the reference points used between them.

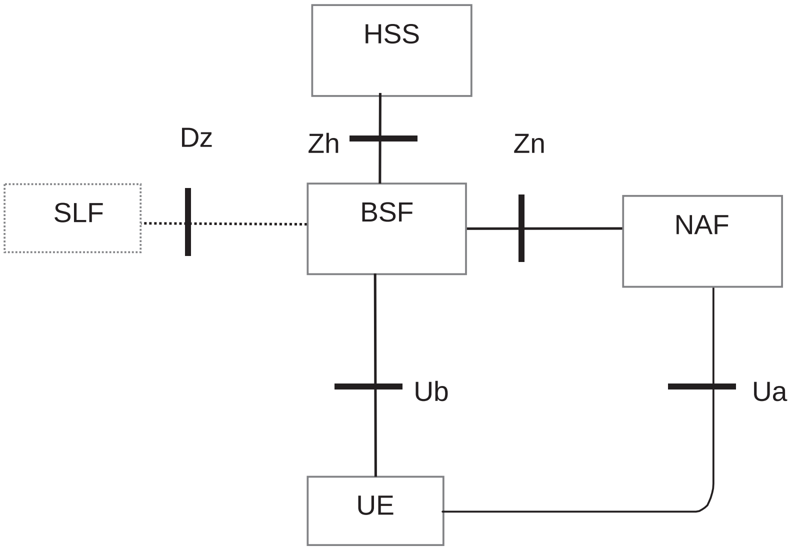
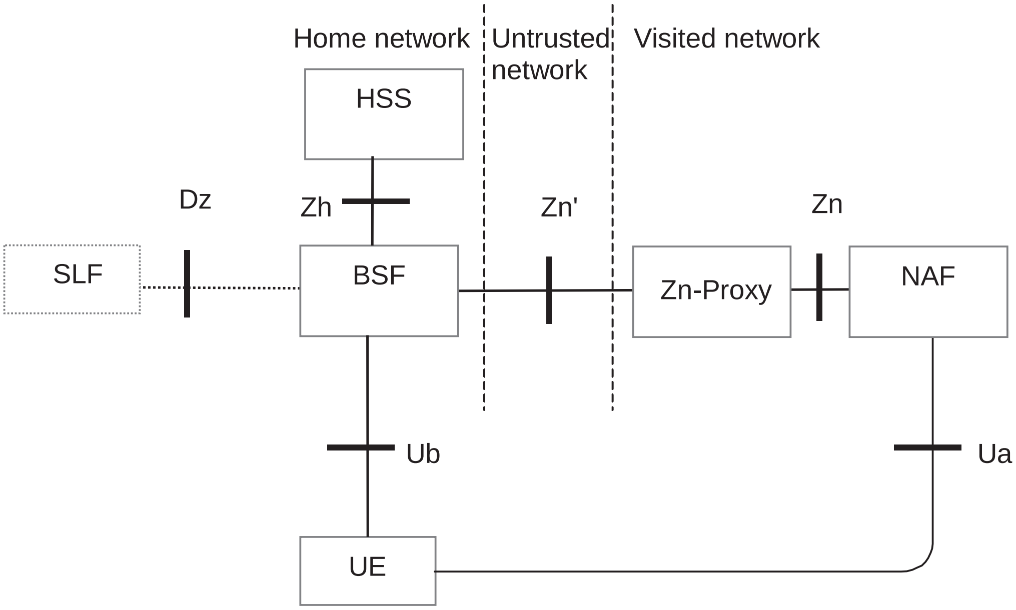


Figure 4.1: Simple network model for bootstrapping involving HSS with Zh reference point

Figure 4.1a shows a simple network model of the entities involved when the network application function is located in the visited network.



NOTE: The Zn' reference point is distinguished from the Zn reference point in that it is used between operators.

Figure 4.1a: Simple network model for bootstrapping in visited network involving HSS with Zh reference point

Figure 4.1b shows a simple network model of the entities involved in the bootstrapping approach when either an HLR or an HSS without Zh reference point support is deployed, and the reference points used between them. The reference point Zh' is optional for the BSF to support.

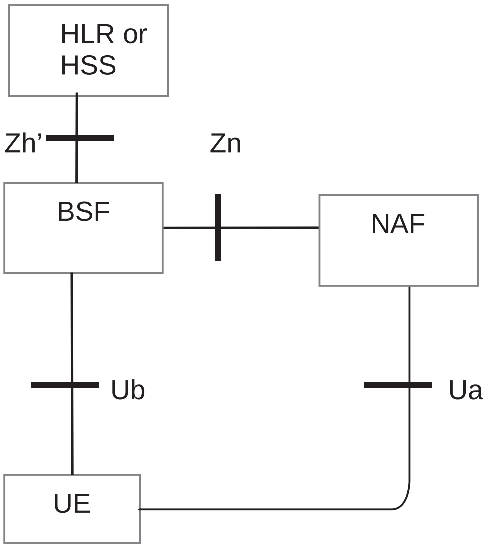


Figure 4.1b: Simple network model for bootstrapping involving either an HLR or an HSS without Zh reference point support

4.2 Network elements

4.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the AKA protocol, and agree on session keys that are afterwards applied between UE and a Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using the key derivation procedure as specified in Annex B. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause 4.5.2.

The BSF shall be able to acquire the GBA user security settings (GUSS) from the HSS.

The BSF shall be able to keep a list, which assigns NAFs to NAF Groups. This list is used to select if any and which application-specific USS within GUSS is valid for a certain NAF.

NOTE 1: The operator does the assignment of NAFs to NAF Groups. NAF Group definitions in HSS and all connected BSFs belonging to the same operator's network shall be equal (cf., clause 4.2.3). As these network elements belong to the same operator's network, standardisation of the NAF Group definitions themselves is not necessary in 3GPP.

NOTE 2: The NAF grouping may be e.g. "home" and "visited". It allows the BSF to send USSs for the same application with e.g. different authorization flags to different NAFs, e.g., in home network and visited networks. The NAF e.g. in visited network indicates only the requested application, but it is unaware of the grouping in home network of the subscriber.

NOTE 3: If support of GBA User Security Settings (GUSS) for service differentiation or GBA_U is desired in combination with HLR or HSS without Zh reference point support, then this can be achieved, for instance by storing the GUSS information in a BSF database (external and/or external to the node itself), or in any other network database which is deemed as appropriate for a specific deployment. GUSS information is not sent over Zh' reference point.

If an HLR or an HSS without Zh reference point support is used within the GBA architecture, then the BSF needs to be configured to use the Zh' reference point with that HLR or HSS. If the Zh reference point is available in the HSS and the full migration has happened, then it shall be used between the BSF and the HSS.

NOTE 4: If an operator wants to upgrade from a GBA architecture using HLR or HSS without Zh reference point support, to one using HSS with Zh reference point support, then the BSF needs to be configured accordingly to use then the Zh reference point. This can also involve a configuration, where gradual replacement is needed. If GBA is deployed from the beginning with an HSS with Zh reference point support then this kind of configuration is not needed.

NOTE 5: During migration from HLR to HSS, the BSF will need to select for a subscriber between HSS and HLR's. Such a mechanism (e.g. configuration based) will not be standardized.

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and a NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of a NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire zero or more application-specific USSs from the HSS via the BSF;
- NAF shall be able to set the local validity condition of the shared key material according to the local policy;

- in the case of GBA_U, the NAF shall be able to determine which key (i.e., Ks_ext_NAF or Ks_int_NAF or both) should be used by using a local policy in the NAF or a key selection indication in the application-specific USS. If the NAF has received an application-specific USS, which contains the key selection indication, this shall override the local policy in the NAF;
- NAF shall be able to check lifetime and local validity condition of the shared key material.

NOTE: Without additional measures, GBA does not guarantee the freshness of the key, Ks(_int/ext)_NAF in the sense that it does not guarantee that the key was not used in a previous run of the Ua protocol. The additional measures which may be taken by the UE and the NAF to ensure key freshness in GBA are:

1) enforce a new run of the Ub protocol (thus generating a new Ks) before deriving a new Ks_NAF.
2) store previously used keys Ks(_int/ext)_NAF, or the corresponding key identifiers B-TID, until the end of their lifetime.

A UE and a NAF that support a Ua protocol that does not provide replay protection over unconnected runs of the protocol, will need to take corresponding action to avoid replay attacks if desired.

4.2.2a Zn-Proxy

In the case where UE has contacted a NAF that is operated in another network than home network, this visited NAF shall use a Zn-Proxy of the NAFs network to communicate with subscriber's BSF (i.e. home BSF).

NOTE: Zn-Proxy functionality may be implemented as a separate network element, or be part of any NE in the visited network that implements Diameter/HTTP proxy functionality (examples of such NE's are the BSF of the network that the visited NAF belongs to, or an AAA-server, or an HTTP server).

General requirements for the functionality of Zn-Proxy are:

- Zn-Proxy shall be able to function as a proxy between the visited NAF, and the subscriber's home BSF;
- Zn-Proxy shall be able to locate subscriber's home BSF and communicate with it over secure channel;
- Zn-Proxy shall be able to validate that the visited NAF is authorized to participate in GBA and shall be able to assert to subscriber's home BSF the visited NAFs DNS name. The Zn-Proxy shall also be able to assert to the BSF that the visited NAF is authorized to request the GBA specific user profiles contained in the NAF request;
- the physical security level of the Zn-proxy shall not be lower than the highest level of the NAFs which it interfaces with.

4.2.3 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS. In the case where the subscriber has multiple subscriptions, i.e. multiple ISIM or USIM applications on the UICC, the HSS may contain one or more GUSSs that can be mapped to one or more private identities, i.e. IMPIs and IMSIs. Each of the existing GUSSs shall be mapped to one or more private identities, but each private identity shall only have zero or one GUSS mapped to it.

The requirements on the HSS are:

- HSS shall provide the only persistent storage for GUSSs;
- GUSS shall be defined in such a way that interworking of different operators for standardised application profiles is possible;
- GUSS shall be defined in such a way that profiles for operator specific applications and extensions to existing application profiles are supported without need for standardisation of these elements.
- GUSS shall be able to contain application-specific USSs that contain parameters that are related to key selection indication in the case of GBA_U (i.e., whether the NAF shall use Ks_ext_NAF or Ks_int_NAF), identification or authorization information of one or more applications hosted by one ore more NAFs. Any other types of parameters are not allowed in the application-specific USS.

NOTE 1: The necessary subscriber profile data may be fetched by the NAF from its local database without involvement with the HSS.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - the type of the UICC the subscriber is issued (i.e. is it GBA_U aware or not, cf. subclause 5);
  - subscriber specific key lifetime:
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or subscriber does not have GUSS then the BSF will use the default values in the BSF local policy defined by the particular MNO.

- HSS shall be able to assign application-specific USSs to a NAF Group. This shall be defined in such a way that different USSs for the same application, but for different groups of NAFs, are possible. The restrictions on the number of USSs per GUSS are dependent on the usage of NAF Groups by the operator:
  - if no NAF Groups are defined for this application then at most one USS per application is stored in GUSS;
  - if NAF Groups are defined for this application then at most one USS per application and NAF Group is stored in GUSS.
- NAF Group definitions in the HSS and all connected BSFs belonging to the same operator's network shall be equal.

4.2.4 UE

The required functionalities from the UE are:

- the support of HTTP Digest AKA protocol;
- the capability to use both a USIM and an ISIM in bootstrapping;
- the capability to select either a USIM or an ISIM to be used in bootstrapping, when both of them are present;
- the capability for a Ua application on the ME to indicate to the GBA Function on the ME the type or the name of UICC application to use in bootstrapping (see clause 4.4.8);
- the capability to derive new key material to be used with the protocol over Ua interface from CK and IK;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]).

A GBA-aware ME shall support both GBA_U, as specified in clause 5.2.1 and GBA_ME procedures, as specified in clause 4.5.

4.2.5 SLF

The SLF:

- is queried by the BSF in conjunction with the Zh interface operation to get the name of the HSS containing the required subscriber specific data.
- is accessed via the Dz interface by the BSF.

The SLF is not required in a single HSS environment. Use of SLF is not required when BSF are configured/managed to use pre-defined HSS.

4.2.6 HLR

If a HLR is used, then the requirement on the HLR is:

- The HLR shall support the request from the BSF for the required authentication vector.

4.3 Bootstrapping architecture and reference points

4.3.1 Reference point Ub

The reference point Ub is between the UE and the BSF. Reference point Ub provides mutual authentication between the UE and the BSF. It allows the UE to bootstrap the session keys based on 3GPP AKA infrastructure.

The HTTP Digest AKA protocol, which is specified in RFC 3310 [4], is used on the reference point Ub. It is based on the 3GPP AKA TS 33.102 [2] protocol. The interface to the USIM is as specified in TS 31.102 [1] and to the ISIM is as specified in TS 31.103 [10].

4.3.2 Reference point Ua

The reference point Ua carries the application protocol, which is secured using the keys material agreed between UE and BSF as a result of the run of HTTP Digest AKA over reference point Ub. For instance, in the case of support for subscriber certificates TS 33.221 [5], it is a protocol, which allows the user to request certificates from the NAF. In this case the NAF would be the PKI portal.

4.3.3 Reference point Zh

The reference point Zh used between the BSF and the HSS allows the BSF to fetch the required authentication information and all GBA user security settings from the HSS. The reference point Zh is an intra-operator domain interface. The interface to the 3G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

4.3.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous HTTP Digest AKA protocol run over the reference point Ub from the UE to the BSF. It is also used to fetch application-specific user security settings from the BSF, if requested by the NAF.

4.3.5 Reference point Dz

The reference point Dz used between the BSF and the SLF allows the BSF to get the name of the HSS containing the required subscriber specific data.

4.3.6 Reference point Zh'

The reference point Zh' used between the BSF and the HLR allows the BSF to fetch the required authentication information. The reference point Zh' is an intra-operator domain interface.

4.4 Requirements and principles for bootstrapping

The following requirements and principles are applicable to bootstrapping procedure:

- the bootstrapping function shall not depend on the particular NAF;
- the server implementing the bootstrapping function needs to be trusted by the home operator to handle authentication vectors;
- the server implementing the NAF needs only to be trusted by the home operator to handle derived key material;

- it shall be possible to support NAF in the operator's home network and in the visited network;
- the architecture shall not preclude the support of network application function in a third network;
- to the extent possible, existing protocols and infrastructure should be reused;
- in order to ensure wide applicability, all involved protocols are preferred to run over IP;
- it shall be prevented that a security breach in one NAF who is using the GBA, can be used by an attacker to mount successful attacks to the other NAFs using the GBA.
- an attacker shall not be able to exploit a security breach in one security protocol over Ua in order to mount a successful attack against a different security protocol over Ua.

4.4.1 Access Independence

Bootstrapping procedure is access independent. Bootstrapping procedure requires IP connectivity from UE.

4.4.2 Authentication methods

Authentication between the UE and the BSF shall not be possible without a valid cellular subscription. Authentication shall be based on the 3GPP AKA protocol.

4.4.3 Roaming

The requirements on roaming are:

- The roaming subscriber shall be able to utilize the bootstrapping function in the home network. The subscriber shall be able to utilize network application function that is in a visited network.
- The home network shall be able to control whether its subscriber is authorized to use the service in the visited network.

4.4.4 Requirements on reference point Ub

The requirements for reference point Ub are:

- the BSF shall be able to identify the UE;
- the BSF and the UE shall be able to authenticate each other based on AKA;
- the BSF shall be able to send a bootstrapping transaction identifier to the UE;
- the UE and the BSF shall establish shared keys;
- the BSF shall be able to indicate to the UE the lifetime of the key material. The key lifetime sent by the BSF over Ub shall indicate the expiry time of the key.

NOTE 1: This does not preclude a UE to refresh the key before the expiry time according to the UE's local policy.

- the BSF and the UE shall protect the permanent user identity IMPI against passive eavesdropping attacks by using a temporary identity. The support of the temporary identity by UE or BSF shall not preclude a successful bootstrapping procedure if the other entity conforms to an earlier release of this specification and does not support the use of a temporary identity.

NOTE 2: User identity privacy can be achieved only when both, UE and BSF, support the use of a temporary identity.

NOTE 3: The use of a temporary identity is not required for 2G GBA (cf. Annex I) as the IMPI is already protected by the mandatory TLS tunnel.

4.4.5 Requirements on reference point Zh

The requirements for reference point Zh are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: This requirement may be fulfilled by physical or proprietary security measures since BSF and HSS are located within the same operator's network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;
- optionally the BSF may have the capability able to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);
- the HSS shall be able to send one 3GPP AKA vector at a time to the BSF;
- the HSS shall be able to send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

NOTE 2: If subscriber's GUSS is updated in HSS, this is not propagated to the BSF. The GUSS in the BSF is updated when the BSF next time fetches the authentication vectors and GUSS from the HSS over Zh reference point as part of the bootstrapping procedure.

- no state information concerning bootstrapping shall be required in the HSS;
- all procedures over reference point Zh shall be initiated by the BSF;
- the number of different interfaces to HSS should be minimized.

4.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;
- If the BSF and the NAF are located within the same operator's network, the DIAMETER based Zn reference point shall be secured according to NDS/IP [13] or may be secured using TLS as specified in Annex E of the present document;
- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E of the present document;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E of the present document;
- The BSF shall verify that the requesting NAF is authorised to obtain the key material or the key material and the requested USS;
- The NAF shall be able to send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall be able to verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The BSF shall be able to send the requested key material to the NAF;
- The NAF shall be able to get a selected set of application-specific USSs from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall be able to indicate to the BSF the single application or several applications it requires USSs for;

NOTE 2: If some application needs only a subset of an application-specific USS, e.g. only one IMPU or MSISDN, the NAF selects this subset from the complete set of USS sent from BSF.

- The BSF shall be able to be configured on a per NAF or per application basis
- whether private subscriber identity, i.e. IMPI, may be sent to the NAF;

- whether a particular USS may be sent to a NAF;

NOTE 3: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI. If HLR is utilized instead of HSS, BSF can be configured to send MSISDN over Zn (but then there is no user anonymity). If the BSF does not send the IMPI, MSISDN or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- If a NAF requests USSs from the BSF and they are not present in subscriber's GUSS, it shall not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF shall then send only the requested and found USSs to the NAF;

- It shall be possible to configure a local policy as follows: BSF may require one or more application-specific USS to be present in a particular subscriber's GUSS for a particular requesting NAF, and to reject the request from the NAF in case the conditions are not fulfilled. In order to satisfy this local policy, it is not required that the NAF requests the USSs over the Zn reference point, which the BSF requires to be present in the GUSS, rather it is sufficient that the BSF checks the presence of the USSs locally. It shall also be possible to configure the BSF in such a way that no USS is required for the requesting NAF;

NOTE 4: For more information on the local policy usage, see Annex J.

- The BSF shall be able to indicate to the NAF the bootstrapping time and the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 5: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 6: If one or more of the USSs that have been delivered to the NAF has been updated in subscriber's GUSS in the HSS, this change is propagated to the NAF the next time it fetches the USS from the BSF over Zn reference point (provided that the BSF has updated subscriber's GUSS from the HSS over Zh reference point).

- The BSF shall remove any existing attribute indicating NAF Grouping from the USSs sent to NAFs.

- NAF shall be able to indicate to BSF the protocol identifier of Ua security protocol it requires the key material by sending NAF-Id to BSF (cf. Annex H).

4.4.7 Requirements on Bootstrapping Transaction Identifier

Bootstrapping transaction identifier (B-TID) shall be used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

Requirements for B-TID are:

- B-TID shall be globally unique;

- B-TID shall be usable as a key identifier in protocols used in the reference point Ua;

- NAF shall be able to detect the home network and the BSF of the UE from the B-TID.

NOTE 1: NAF can remove the security association based on deletion conditions after the key has become invalid.

NOTE 2: Care has to be taken that the parallel use of GBA and non-GBA authentication between UE and NAF does not lead to conflicts, e.g. in the name space. This potential conflict cannot be resolved in a generic way as it is dependent on specific protocol and authentication mechanism used between UE and application server. It is therefore out of scope of this specification.
For the example of HTTP Digest authentication used between UE and NAF, parallel use is possible as the following applies: <username,password>-pairs must be unique to one realm only. As the NAF controls the realm names, it has to ensure that only the GBA based realm is named with the reserved 3GPP realm name. In the special case that the NAF wants to allow non GBA based authentication in the GBA realm also, it has to ensure that no usernames in the format of a B-TID are used outside GBA based authentication.

4.4.8 Requirements on selection of UICC application and related keys

When several applications are present on the UICC, which are capable of running AKA, then the ME shall choose one of these UICC applications for performing the GBA procedures specified in this document in the following order of preference:

1. The UE determines which UICC application is to be involved:

   a. the application on the ME that needs Ks_NAF (Ua application) may indicate to the GBA support function (GBA function) the type or the name of the UICC application: no preference, USIM, ISIM, or the "Label" (see definition in TS 31.101 [15]) of the UICC application.

NOTE 1: A Ua application specification may require the use of only a USIM (e.g. in MBMS) or only an ISIM.

NOTE 2: A user or operator may want to use a Ua application with a specific UICC application indicated by the 'Label'. This could be configured in the Ua application in the ME by the user or the operator.

A Ua application may require to use the same UICC application in the first and all consecutive runs of Ub protocol for a Ua application instance to ensure that IMPI is not changed during a Ua application session which lasts over several runs of Ub protocol. In this case the Ua application shall request the GBA function to run the Ub protocol with the UICC application that is indicated by the corresponding "Label" or IMPI, depending on which one is available. If both are available, then IMPI shall be used to indicate which UICC application is to be used by the GBA function.

If the application on the ME indicated a "Label" of the UICC application, step b below shall be executed.

If the application on the ME indicated that the UICC application type should be:

- the USIM; step b below is skipped and in steps c and d only USIM applications are considered.
- the ISIM; step b below is skipped and in steps c and d only ISIM applications are considered.

  if the application on the ME did not indicate a preference, step b below is skipped and the selection process is executed as described below, starting with step c;

   b. if a "Label" was indicated in step a:

At most, there can be only one USIM active at one time. Therefore, if the USIM indicated in the "Label" by the Ua application is different to the currently active USIM application, then the ME shall reject this request.

If a different ISIM to the currently active ISIM application(s) is indicated to the GBA support function by the Ua application, then the ME shall not terminate the currently active ISIM application(s) but the ME shall follow the procedure in chapter 4.4.8.1 when activating the ISIM application indicated by the "Label", as the UE is allowed to have several ISIM's active simultaneously.

   c. if no "Label" was indicated in step a and there are UICC applications active:

If a preferred UICC application type was indicated but no UICC application of this type is active then step d shall be followed.

If a preferred UICC application type was indicated and there are active UICC applications of this preferred type, then the GBA function shall choose:

  - if the preferred UICC application type is USIM then the active USIM is selected

Page 200 of 1361.

- if the preferred UICC application type is ISIM and only one ISIM is active then this is selected

- if the preferred UICC application type is ISIM and more than one ISIM is active then the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of all active ISIM applications on the UICC), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select an active ISIM.

If no preference was given and there is more than one active UICC application, the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of all active UICC applications), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select the active USIM application, if an active USIM application exists, otherwise any active ISIM application.

If no preference was given and there is only one active UICC application, then the GBA function selects this active UICC application;

d. if no "Label" was indicated in step a and if there are no UICC applications active active or if there is no UICC application of the preferred UICC application type active:

- if there is only one UICC application on the UICC, the GBA function selects it, if possible;

- if there is more than one UICC application on the UICC, the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user chooses the UICC application to be selected. If a preferred UICC application type was indicated and there are UICC applications of this type on the UICC, then the list shown contains only UICC applications of this type, otherwise the list contains all UICC applications on the UICC. If no dialogue is shown the GBA function shall select the "last selected" UICC application of the preferred type (i.e. either the "last selected" USIM or the "last selected" ISIM depending on the given preference), if possible. In case the Ua application indicated "no preference" and both USIM and ISIM are present on the UICC, then the "last selected" USIM is selected.

The procedure in clause 4.4.8.1 shall be followed.

e. if the UICC application type indicated in step a and used in step c and/or d was ISIM, but there was no ISIM to select, then step c and/or d is repeated with UICC application type USIM; otherwise the selection process fails.

NOTE 3: Step e is required for the case that an ISIM as defined in TS 33.203 [16] may be realised using a USIM application on the UICC.

2. If there already is a key Ks derived from the chosen UICC application, the UE takes this key to derive Ks_NAF.

3. If there is no such key Ks, the UE first runs the Ub protocol involving the selected UICC application and then goes to step 2.

If a USIM is chosen, the IMPI obtained from the IMSI stored on the USIM as specified in TS 23.003 [11] clause 13.3, is used in the protocol run over Ub.

NOTE 4: Strictly speaking, an IMPI, and the derivation of an IMPI from an IMSI as in TS 23.003 [11], clause 13 are only defined in the context of the IMS. For the purposes of this specification, however, an identifier obtained from an IMSI as specified in TS 23.003 [11], clause 13.3 is also called an IMPI, even if the user has no IMS subscription.

If an ISIM is selected, the IMPI stored on the ISIM is used in the protocol run over Ub.

Whenever a UICC application is successfully selected or terminated, the rules in this clause for choosing the UICC application are re-applied and, consequently, the UICC application chosen for GBA may change.

NOTE 5: At any one time, there is at most one UICC application chosen for performing the GBA procedures.

4.4.8.1 UICC application activation procedure in GBA

UICC application activation is defined in TS 31.101 [15].

NOTE: As part of the UICC application (USIM or ISIM) activation procedure, the UICC may require user verification e.g. PIN entry.

If activation of a new UICC application fails then the GBA function shall indicate this to the Ua application.

4.4.9 Requirements on reference point Ua

The generic requirements for reference point Ua are:

- the UE and the NAF shall be able to secure the reference point Ua using the GBA-based shared secret;

NOTE: The exact method of securing the reference point Ua depends on the application protocol used over reference point Ua.

- in the case of GBA_U, the UE and the NAF shall be able to agree which key (i.e, Ks_ext_NAF or Ks_int_NAF or both) is used as the GBA-based shared secret if both keys may be used;

There are two ways to have an agreement between the UE and the NAF which key shall be used Ks_(ext)_NAF or Ks_int_NAF or both:

a) In a generic case, where the protocol used over reference point Ua can be used for different applications (e.g., HTTPS), the protocol should be able to indicate which key should be used.

b) In a specific case, where the protocol is application specific (e.g., MIKEY in MBMS), the agreement can be based on implicit knowledge.

- any security protocol over Ua shall be associated with a Ua security protocol identifier. This identifier shall be specified in Annex H of this specification.
- the NAF shall be able to indicate to the UE that GBA-based shared secret should be used;
- the NAF shall be able to indicate to the UE that the current shared secret has expired and the UE should use newer shared secret with the NAF.
- The default lifetime of the NAF specific key material Ks_(ext/int)_NAF shall be equal to the lifetime of Ks when not specified within the Ua-application specification. The lifetime of the Ks_(ext/int)_NAF shall not exceed the lifetime of corresponding Ks. If a lifetime for the Ks_(ext/int)_NAF (or further adapted key material) is available in the NAF, due to a Ua application specification having its own lifetime value or due to NAF having it's own policy for the adapted key material, then if this lifetime is different from the Ks lifetime received from the BSF, then the NAF shall always select the minimum value for the lifetime out of these two.
- The UE and NAF may adapt the key material Ks_(ext/int)_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification. The default lifetime of the adapted key material shall be equal to the lifetime of Ks_(ext/int)_NAF when not specified within the Ua-application specification. The lifetime of the adapted key material shall not exceed the lifetime of corresponding Ks_(ext/int)_NAF. If a lifetime for the Ks_(ext/int)_NAF (or further adapted key material) is available in the NAF, due to a Ua application specification having its own lifetime value or due to NAF having it's own policy for the adapted key material, then if this lifetime is different from the Ks lifetime received from the BSF, then the NAF shall always select the minimum value for the lifetime out of these two.

4.4.10 Requirements on reference point Dz

This interface between BSF and SLF is used to retrieve the address of the HSS which holds the subscription for a given user. This interface is not required in a single HSS environment.

4.4.11 Requirements on GBA keys and parameters handling

When referring to GBA keys, the following keys are intended: Ks and NAF specific keys derived from the Ks. When referring to NAF specific keys, the following keys are intended: Ks_ext/int_NAF (in GBA_U context) and Ks_NAF (in GBA_ME context), and any keys derived from these keys. The notation Ks_(ext/int)_NAF refers to Ks_ext/int_NAF in GBA_U context and Ks_NAF in GBA_ME context. The notation Ks_(ext)_NAF refers to Ks_ext_NAF in GBA_U context, and Ks_NAF in GBA_ME context.

The ME shall delete all GBA keys (i.e., Ks, and NAF specific keys) and the corresponding NAF_IDs, B-TID, Ks_(int/ext)_NAF lifetimes, Ks lifetime, and lifetime (of the keys derived from Ks_(ext)_NAF) when at least one of the conditions below is met:

1 the UICC is removed from the ME when the ME is in power on state;

2 the ME is powered up and the ME discovers that another UICC has been inserted to the ME. For this, the ME needs to store in non-volatile memory the last inserted UICC-identity to be able to compare that with the used UICC-identity at UICC insertion and power up; or

3 the ME is powered up and the ME discovers that no UICC has been inserted to the ME.

NOTE 1: One possible way, how this requirement can be fulfilled by an application in an open platform is, if the keys are deleted at shut-down and at start-up of the application. When the ME operating system detects one of the conditions above, it can shut down the application to force key deletion. The deletion at start-up ensures that keys are also deleted, when an irregular power-down or UICC removal during power down has occurred.

The ME shall delete all GBA keys related to a certain Ks (i.e., Ks itself, and NAF specific keys derived from this specific Ks) and the corresponding NAF_IDs, B-TID, Ks_(ext/int)_NAF lifetimes, Ks lifetime, and lifetime (of the keys derived from Ks_(ext)_NAF) when the key lifetime of this specific Ks expires.

In the case of GBA_ME, the key Ks shall be deleted from the ME when the ME is powered down. The NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any) may be deleted from the ME when the ME is powered down. If the ME does not delete these NAF specific keys at power down then the NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any) together with the NAF_IDs, B-TID, Ks_(ext)_NAF lifetime and lifetimes (of the keys derived from Ks_(ext)_NAF) shall be stored in non-volatile memory.

If the NAF specific keys are stored in non-volatile memory, then when the ME is powered up again, the ME may need to ensure that the same UICC application is selected for the Ua application, in order to allow the re-use of the NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any), cf. clause 4.4.8. For this, the ME shall store also the IMPI in non-volatile memory. If the same UICC application can not be selected for a Ua application at UE power up, then the ME shall delete the NAF specific keys related to that IMPI stored in non-volatile memory.

Whenever a UICC application is terminated (see section 4.4.8) the shared key Ks established from it in the protocol over the Ub reference point (according to clauses 4.5.2 and 5.3.2) shall be deleted.

NOTE 2: In case the key Ks has been deleted, but the same UICC is still present (i.e. none of conditions 1, 2 or 3 is met), the Ua applications can continue using the NAF specific keys (Ks_(ext/int)_NAF) until the Ks lifetime expires.

4.4.12 Requirements on reference point Zh'

This reference point is optional for the BSF to support. The requirements for reference point Zh' are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: This requirement may be fulfilled by physical or proprietary security measures, since BSF and HLR are located within the same operator's network.

- the BSF shall be able to send an authentication vector request concerning a subscriber;
- the HLR shall be able to send one authentication vector, as described in TS 29.109 [32] at a time to the BSF;
- no other GBA functionality than conveying authentication vectors shall be required on Zh';
- no state information concerning bootstrapping shall be required in the HLR;
- all procedures over reference point Zh' shall be initiated by the BSF;
- the number of different interfaces to HLR should be minimized.

NOTE 2: If support of GBA User Security Settings (GUSS) is desired in combination with HLR or HSS with Zh' reference point support, then this can be achieved, for instance by storing the GUSS information in a BSF database (external and/or external to the node itself), or in any other network database which is deemed as appropriate for a specific deployment. GUSS information is not sent over Zh' reference point.

4.4.13 Requirements on TMPI handling

The BSF shall store a TMPI together with the IMPI, from which it was derived (cf. Annex B.4), until the next bootstrapping procedure is executed using this TMPI.

The BSF may have a local policy for deleting stored (TMPI, IMPI)-pairs before the next bootstrapping procedure is executed using this TMPI, e.g. for storage or performance reasons.

The ME shall store a TMPI together with the IMPI, from which it was derived (cf. Annex B.4), in non-volatile memory.

The ME shall delete all stored (TMPI, IMPI)-pairs when at least one of the conditions below is met:

1. the UICC is removed from the ME when the ME is in power on state; or

2. the ME is powered up and the ME discovers that another UICC has been inserted to the ME. For this, the ME needs to store in non-volatile memory the last inserted UICC-identity to be able to compare that with the used UICC-identity at UICC insertion and power up; or

3. the ME is powered up and the ME discovers that no UICC has been inserted to the ME.

4.5 Procedures

This chapter specifies in detail the format of the bootstrapping procedure that is further utilized by various applications. It contains the AKA authentication procedure with BSF, and the key material generation procedure.

4.5.1 Initiation of bootstrapping

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use the GBA. When a UE wants to interact with a NAF, but it does not know if the NAF requires the use of shared keys obtained by means of the GBA, the UE may contact the NAF for further instructions (see figure 4.2).

NOTE: The above text implies that a UE may contact either the BSF or the NAF without knowing whether the NAF supports GBA

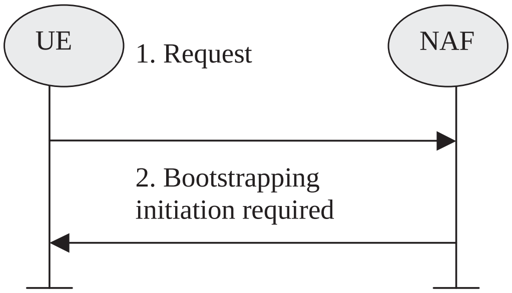


Figure 4.2: Initiation of bootstrapping

1. The UE may start communication over reference point Ua with the NAF with or without any GBA-related parameters.

2. If the NAF requires the use of shared keys obtained by means of the GBA, but the request from UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message. The form of this initiation message may depend on the particular reference point Ua.

4.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 4.3). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. subclause 4.5.3).

NOTE 1: The main steps from the specifications of the AKA protocol in TS 33.102 [2] and the HTTP digest AKA protocol in RFC 3310 [4] are repeated in figure 3 for the convenience of the reader. In case of any potential conflict, the specifications in TS 33.102 [2] and RFC 3310 [4] take precedence.

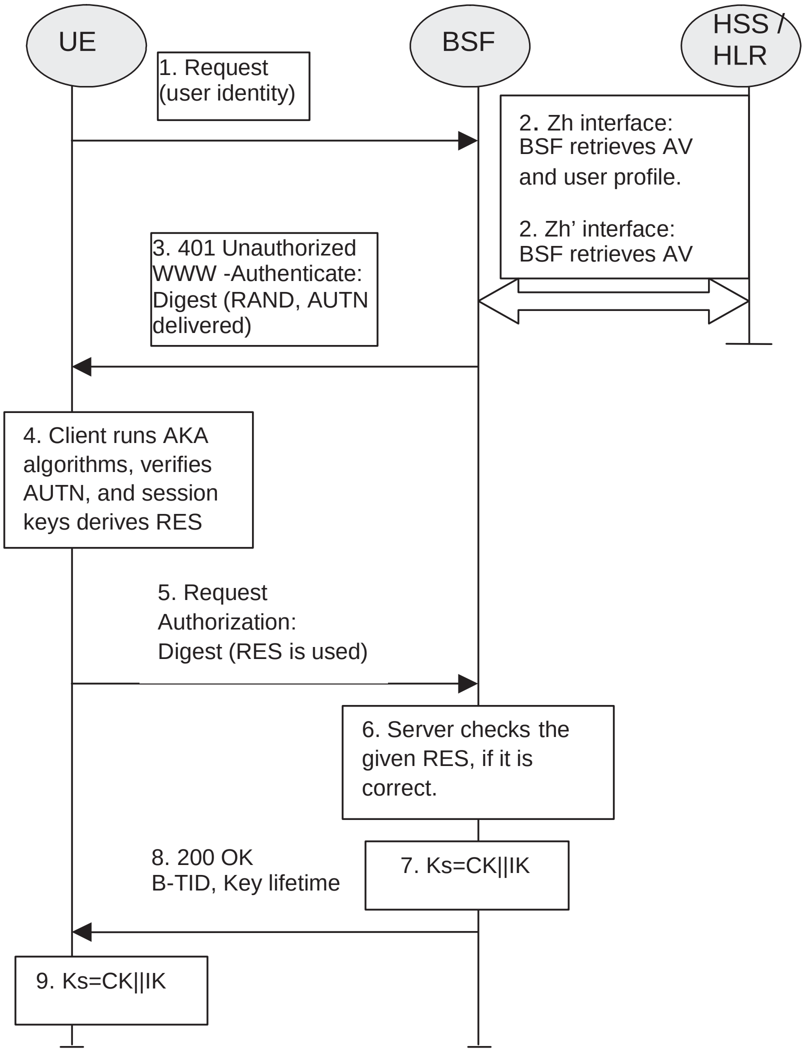

UE
BSF
HSS / HLR
1. Request (user identity)
2. Zh interface: BSF retrieves AV and user profile.
2. Zh' interface: BSF retrieves AV
3. 401 Unauthorized WWW -Authenticate: Digest (RAND, AUTN delivered)
4. Client runs AKA algorithms, verifies AUTN, and session keys derives RES
5. Request Authorization: Digest (RES is used)
6. Server checks the given RES, if it is correct.
7. Ks=CK||IK
8. 200 OK B-TID, Key lifetime
9. Ks=CK||IK

Figure 4.3: The bootstrapping procedure

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when communicating over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when communicating over Ub.

NOTE 1a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

1. The UE sends an HTTP request towards the BSF. When a TMPI associated with the IMPI in use is available on the UE, the UE includes this TMPI in the "username" parameter, otherwise the UE includes the IMPI.

2. The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF looks up the corresponding IMPI in its local database. If the BSF does not find an IMPI corresponding to the received TMPI it returns an appropriate error message to the UE. The UE then deletes the TMPI and retries the request using the IMPI.

   The BSF retrieves the complete set of GBA user security settings and one Authentication Vector (AV, AV = RAND||AUTN||XRES||CK||IK) over the reference point Zh from the HSS.

   In the case that no HSS with Zh reference point is deployed, the BSF retrieves the Authentication Vector over the reference point Zh' from either an HLR or an HSS with Zh' reference point support.

   If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

NOTE 2: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 2.

3. Then BSF forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.

4. The UE checks AUTN to verify that the challenge is from an authorised network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both BSF and UE.

5. The UE sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.

6. The BSF authenticates the UE by verifying the Digest AKA response.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

7. The BSF generates key material Ks by concatenating CK and IK. The B-TID value shall be also generated in format of NAI by taking the base64 encoded (cf. RFC 3548 [12]) RAND value from step 3, and the BSF server name, i.e. base64encode(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

   If the request included the product token "3gpp-gba-tmpi" in the user agent request-header field the BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

8. The BSF shall send a 200 OK message, including a B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks. The key material Ks is generated in UE by concatenating CK and IK.

9. Both the UE and the BSF shall use the Ks to derive the key material Ks_NAF during the procedures as specified in clause 4.5.3. Ks_NAF shall be used for securing the reference point Ua.

   Ks_NAF is computed as Ks_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id), where KDF is the key derivation function as specified in Annex B, and the key derivation parameters consist of the user's IMPI, the NAF_Id and RAND. The NAF_Id is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is specified in Annex H. KDF shall be implemented in the ME.

NOTE 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

To allow consistent key derivation based on NAF name in UE and BSF, at least one of the three following prerequisites shall be fulfilled:

(1) The NAF is known in DNS under one domain name (FQDN) only, i.e. no two different domain names point to the IP address of the NAF. This has to be achieved by administrative means.
This prerequisite is not specific to 3GPP, as it is necessary also under other circumstances, e.g. for TLS V1.0 [28] without use of wildcard or multiple-name certificates.

(2) Each DNS entry of the NAF points to a different IP address. The NAF responds to all these IP addresses. Each IP address is tied to the corresponding FQDN by NAF configuration. The NAF can see from the IP address, which FQDN to use for key derivation.

(3) Ua uses a protocol which transfers the host name (FQDN of NAF as used by UE) to NAF (e.g. HTTP/1.1 with mandatory Host request header field). This requires the NAF to check the validity of the host name, to use this name in all communication with UE where appropriate, and to transfer this name to BSF to allow for correct derivation of Ks_NAF.
In case of a TLS tunnel this requires either multiple-identities certificates or the deployment of TLS Extensions as specified in Annex E of TS 33.310 [19] or other protocol means with similar purpose.

The UE and the BSF shall store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated or until the deletion conditions are satisfied (see 4.4.11).

NOTE 5: The following case can occur. The UE contacts the NAF1 and generates keys for NAF1. Then the UE contacts NAF2 and generates NAF2 keys. Then NAF1 requests then keys from the BSF, but the old key keys could have been overwritten due to NAF2 having initiated a new GBA run. The UE initiates a new GBA-run (B-TID2) after handling NAF1 (B-TID1) and starting the request to the NAF1 over Ua. One possible reason is that B-TID1 lifetime was about to expire. It is very likely that the GBA-run takes much more time (HSS involvement) then the Zn/Ua request such that the B-TID1 request at the BSF should arrive in most cases earlier at the BSF. So this out-of-order case should be very rare. This error situation will be signalled back to the UE, such that the most recent B-TID2 will also be used for NAF1. This out-of order case is self-correcting, since if the B-TID1 is unknown in the BSF, then the Ua request will fail and the UE can send a new request using B-TID2.

If the response included the product token "3gpp-gba-tmpi" in the server response-header field the UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

4.5.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 4.5.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with an NAF the following steps are executed as depicted in figure 4.4.

1. UE starts communication over reference point Ua with the NAF:

   - in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:

     - if a key Ks for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 4.5.2;

     - if no key Ks for the selected UICC application is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF

If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF.

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g. because the key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure 4.5. If the key's lifetime has expired the protocol used over reference point Ua shall be terminated. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause 4.5.2, in order to obtain a new key Ks.

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 1: If the shared key between UE and NAF is invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

- the UE supplies the B-TID to the NAF, in the form as specified in clause 4.5.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 2: The UE may adapt the key material Ks_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification.

- the key management procedures for GBA related keys in the ME (i.e. Ks and Ks_NAF keys) are described in section 4.4.11.

- when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

  According to the procedures defined in clauses 4.5.2 and 4.5.3, in the UE there is at most one Ks_NAF key stored per NAF-Id.

2. NAF starts communication over reference point Zn with BSF

   - The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua.;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF-Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

3. The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key Ks_NAF, as well as the bootstrapping time and the lifetime of that key, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

   NOTE 4: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

   NOTE 5: The NAF will adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.
- The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy;

4. NAF continues with the protocol used over the reference point Ua with the UE.

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way.

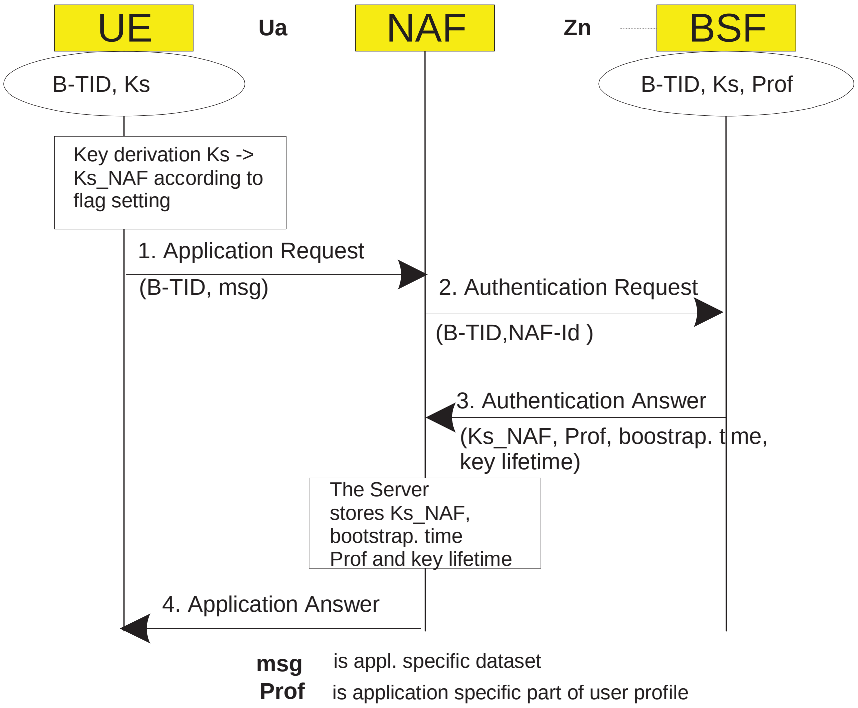

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID,NAF-Id)
3. Authentication Answer
(Ks_NAF, Prof, boostrap. time, key lifetime)
The Server stores Ks_NAF, bootstrap. time Prof and key lifetime
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure 4.4: The bootstrapping usage procedure

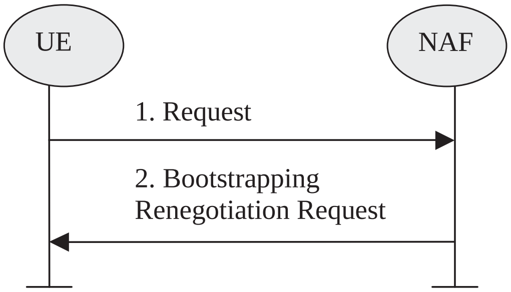


Figure 4.5: Bootstrapping renegotiation request

4.5.4 Procedure related to service discovery

The UE shall discover the address of the BSF the from the identity information related to the UICC application that is used during bootstrapping procedure, i.e., IMSI for USIM, or IMPI for ISIM. The address of the BSF is derived as specified in TS 23.003 [11].

5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U)

It is assumed that the UICC, BSF, and HSS involved in the procedures specified in this clause are capable of handling the GBA_U specific enhancements. The procedures specified in this clause also apply if NAF is not GBA_U aware.

5.1 Architecture and reference points for bootstrapping with UICC-based enhancements

The text from clause 4.4 of this specification applies also here, with the addition that the interface between the ME and the UICC, as specified in TS 31.102 [1] and TS 31.103 [10], needs to be enhanced with GBA_U specific commands. The requirements on these commands can be found in clause 5.2.1, details on the procedures are in clause 5.3.

5.2 Requirements and principles for bootstrapping with UICC-based enhancements

The requirements and principles from clause 4.4 also apply here with the following addition:

5.2.1 Requirements on UE

The 3G AKA keys CK and IK resulting from a run of the protocol over the Ub reference point shall not leave the UICC.

The UICC shall be able to distinguish between authentication requests for GBA_U, and authentication requests for other 3G authentication domains.

Upon an authentication request from the ME, which the UICC recognises as related to GBA_U, the UICC shall derive the bootstrapping key.

Upon request from the ME, the UICC shall be able to derive further NAF-specific keys from the derived key stored on the UICC.

All GBA-aware MEs shall support procedures for the two previous requests.

5.2.2 Requirements on BSF

BSF shall support both GBA_U and GBA_ME bootstrapping procedures. The decision on running one or the other shall be based on subscription information (i.e. UICC capabilities).

The BSF shall be able to acquire the UICC capabilities related to GBA as part of the GBA user security settings received from the HSS.

5.3 Procedures for bootstrapping with UICC-based enhancements

5.3.1 Initiation of bootstrapping

The text from clause 4.5.1 of the present document applies also here.

5.3.2 Bootstrapping procedure

The procedure specified in this clause differs from the procedure specified clause 4.5.2 in the local handling of keys and Authentication Vectors in the UE and the BSF. The messages exchanged over the Ub reference point are identical for both procedures.

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 5.1). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping renegotiation indication from the NAF, or when the lifetime of the key in UE has expired (see clause 5.3.3).

NOTE: The main steps from the specifications of the AKA protocol in TS 33.102 [2] and the HTTP digest AKA protocol in RFC 3310 [4] are repeated in figure 5.1 for the convenience of the reader. In case of any potential conflict, the specifications in TS 33.102 [2] and RFC 3310 [4] take precedence.

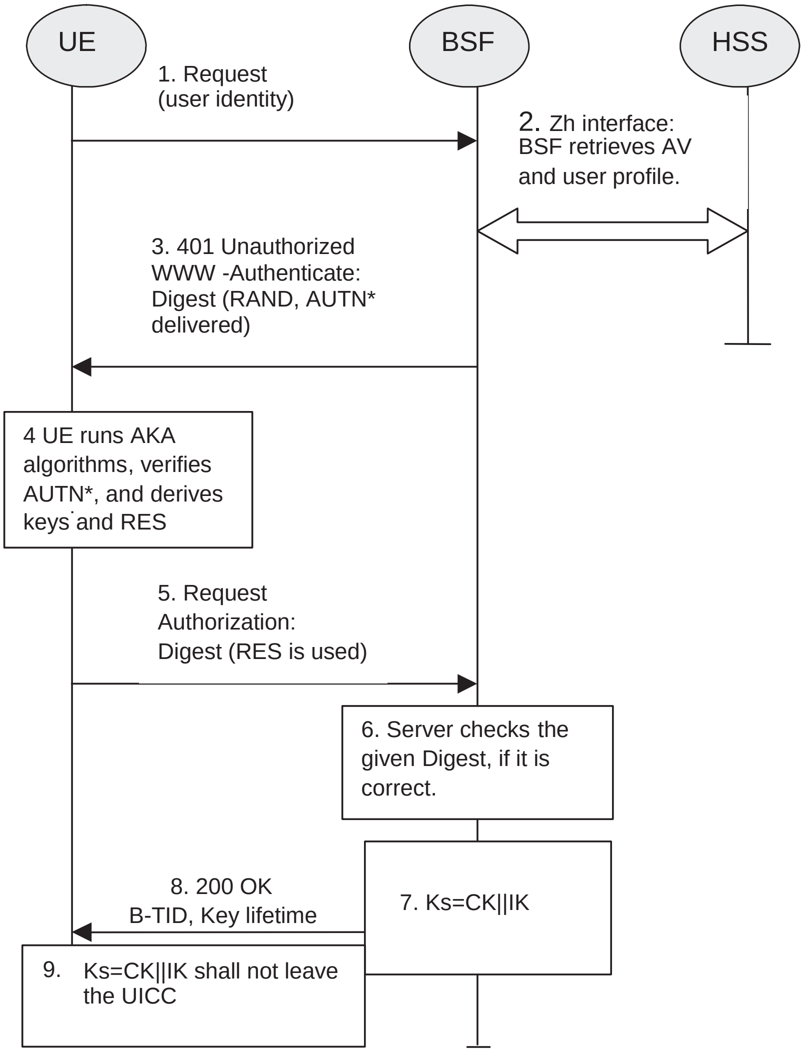

UE
BSF
HSS
1. Request (user identity)
2. Zh interface: BSF retrieves AV and user profile.
3. 401 Unauthorized WWW -Authenticate: Digest (RAND, AUTN* delivered)
4 UE runs AKA algorithms, verifies AUTN*, and derives keys and RES
5. Request Authorization: Digest (RES is used)
6. Server checks the given Digest, if it is correct.
7. Ks=CK||IK
8. 200 OK B-TID, Key lifetime
9. Ks=CK||IK shall not leave the UICC

Figure 5.1: The bootstrapping procedure with UICC-based enhancements

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when communicating over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when communicating over Ub.

NOTE a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

1. The ME sends an HTTP request towards the BSF. When a TMPI associated with the IMPI in use is available on the UE, the UE includes this TMPI in the "username" parameter, otherwise the UE includes the IMPI.

2. The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF looks up the corresponding IMPI in its local database. If the BSF does not

find an IMPI corresponding to the received TMPI it returns an appropriate error message to the UE. The UE then deletes the TMPI and retries the request using the IMPI.

The BSF retrieves the complete set of GBA user security settings and one Authentication Vector (AV, AV = RAND||AUTN||XRES||CK||IK) over the Zh reference point from the HSS.

If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

The BSF can then decide to perform GBA_U, based on the user security settings (USSs). In this case, the BSF proceeds in the following way:

- BSF computes MAC* = MAC⊕ Trunc(SHA-1(IK))

NOTE 1: Trunc denotes that from the 160 bit output of SHA-1 [23], the 64 bits numbered as [0] to [63] are used within the * operation to MAC.

The BSF stores the XRES after flipping the least significant bit.

NOTE 2: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 2.

3. Then BSF forwards the RAND and AUTN* (where AUTN* = SQN ⊕ AK || AMF || MAC*) to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.

4. The ME sends RAND and AUTN* to the UICC. The UICC calculates IK and MAC (by performing MAC= MAC* ⊕ Trunc(SHA-1(IK))). Then the UICC checks AUTN(i.e. SQN ⊕ AK || AMF || MAC) to verify that the challenge is from an authorised network; the UICC also calculates CK and RES. This will result in session keys CK and IK in both BSF and UICC. The UICC then transfers RES (after flipping the least significant bit) to the ME and stores Ks, which is the concatenation of CK and IK, on the UICC.

5. The ME sends another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.

6. The BSF authenticates the UE by verifying the Digest AKA response.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

7. The BSF generates the key Ks by concatenating CK and IK. The B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e. base64encode(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

If the request included the product token "3gpp-gba-tmpi" in the user agent request-header field the BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

8. The BSF shall send a 200 OK message, including the B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks.

9. Both the UICC and the BSF shall use the Ks to derive NAF-specific keys Ks_ext_NAF and Ks_int_NAF during the procedures as specified in clause 5.3.3, if applicable. Ks_ext_NAF and Ks_int_NAF are used for securing the Ua reference point.

Ks_ext_NAF is computed in the UICC as Ks_ext_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_Id), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = KDF(Ks, "gba-u, RAND, IMPI, NAF_Id), where KDF is the key derivation function as specified in Annex B, and the key derivation parameters include the user's IMPI, the NAF_Id and RAND. The NAF_Id is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is specified in Annex H. The key derivation parameters used for Ks_ext_NAF derivation must be different from those used for Ks_int_NAF derivation. This is done by adding a static string "gba-me" in Ks_ext_NAF and "gba-u" in Ks_int_NAF as an input parameter to the key derivation function.

NOTE 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

To allow consistent key derivation based on NAF name in UE and BSF, at least one of the prerequisites which are specified in clause 4.5.2 shall be met.

The UICC and the BSF store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated or until the deletion conditions are satisfied (see 4.4.11).

If the response included the product token "3gpp-gba-tmpi" in the server response-header field the UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

5.3.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 5.3.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with a NAF the following steps are executed as depicted in figure 5.3.

Next, the UE and the NAF have to agree, which type of keys to use, Ks_ext_NAF or Ks_int_NAF, or both. The default is the use of Ks_ext_NAF only. This use is also supported by MEs and NAFs, which are GBA_U unaware. If Ks_int_NAF, or both Ks_ext_NAF and Ks_int_NAF are to be used, this use has to be agreed between UE and NAF prior to the execution of the procedure described in the remainder of this clause 5.3.3. Any such agreement overrules the default use of the keys. A key selection indication, which key (i.e. Ks_int_NAF or Ks_ext_NAF) the NAF shall use in the Ua reference point may be present in the application specific USS as defined in stage 3 specification. If the indication exists, the NAF shall use the indicated key. If the Ks_int_NAF key was indicated in the USS, the UE attempts to use Ks_ext_NAF key, the NAF shall terminate the communication with the UE.

NOTE 1: This agreement may be mandated by the specification, which defines the Ua reference point between UE and NAF, e.g. TS 33.246 for the use of GBA in MBMS, or negotiated by the NAF and the UE over the Ua reference point, or reached by configuration.

1. UE starts communication over reference point Ua with the NAF using the keys Ks_ext_NAF or Ks_int_NAF, or both, as required:

    - in general, UE and NAF will not yet share the key(s) required to protect the Ua reference point. If they do not, the UE proceeds as follows:

- if Ks_ext_NAF is required and a key Ks for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_ext_NAF from Ks, as specified in clause 5.3.2;

- if Ks_int_NAF is required and a key Ks for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_int_NAF from Ks, as specified in clause 5.3.2;

  If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_ext/int_NAF, then the UE should first agree on new key Ks with the BSF over the Ub reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_ext_NAF or Ks_int_NAF, or both, as required.

- if Ks for the selected UICC application is not available in the UE, the UE first agrees on a new key Ks with the BSF over the Ub reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_ext_NAF or Ks_int_NAF, or both, as required;

- if the NAF shares a key with the UE, but the NAF requires an update of that key, it shall send a suitable bootstrapping renegotiation request to the UE. If the key's lifetime has expired the protocol used over reference point Ua shall be terminated. The form of this indication depends on the particular protocol used over Ua reference point. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over Ub, as specified in clause 5.3.2, in order to obtain new keys.

NOTE 2: If the shared keys between UE and NAF become invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

NOTE 3: If it is not desired by the NAF to use the same Ks to derive more than one Ks_int/ext_NAF then the NAF can reply to the first request sent by a UE by sending a key update request to the UE.

- The UE supplies the B-TID to the NAF, as specified in clause 5.3.2, to allow the NAF to retrieve the corresponding keys from the BSF

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (cf. clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 4 The UE may adapt the keys Ks_ext_NAF or Ks_int_NAF to the specific needs of the Ua reference point. This adaptation is outside the scope of this specification.

- the key management procedures for GBA related keys in the ME (i.e. Ks_ext_NAF keys) are described in section 4.4.11.

  - all GBA related keys in the UICC do not need to be deleted when the ME is powered down.

NOTE 5: After each run of the protocol over the Ub reference point, a new key Ks, associated with a new B-TID, are derived in the UE according to clause 5.3.2, so that it can never happen, that key Ks with different B-TIDs simultaneously exist in the UE.

- When new key Ks is agreed over the Ub reference point and new NAF-specific keys need to be derived for one NAF_Id, then both, Ks_ext_NAF and Ks_int_NAF (if present), shall be updated for this NAF_Id, but other keys Ks_ext_NAF or Ks_int_NAF relating to other NAF_Ids, which may be stored on the UE, shall not be affected.

  According to the procedures defined in clauses 5.3.2 and 5.3.3, in the UE there is at most one Ks_int_NAF/Ks_ext_NAF key pair stored per NAF_Id.

NOTE 6: This rule ensures that the keys Ks_ext_NAF and Ks_int_NAF are always in synch at the UE and the NAF.

2. NAF now starts communication over the Zn reference point with the BSF.

- The NAF requests from the BSF the keys corresponding to the B-TID, which was supplied by the UE to the NAF over the Ua reference point. If the NAF is GBA_U aware it indicates this by including a corresponding flag in the request;

- The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

NOTE 7: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

- With the keys request over the Zn reference point, the NAF shall supply a NAF-Id (which includes NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

3. The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF supplies the bootstrapping time and the lifetime time of these keys, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to

the NAF.If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request (See figure 4.5) to the UE;

NOTE 8: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

NOTE 9: The NAF will adapt the keys Ks_ext_NAF and Ks_int_NAF to the specific needs of the Ua reference point in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.

- The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy.

4. The NAF now continues with the protocol used over the Ua reference point with the UE.

- If the NAF requested an application-specific USS from the BSF and the USS was returned the NAF, the NAF shall check whether this USS contains an key selection indication. If the key selection indication is present, the NAF shall use only the indicated key. If a different key was used over Ua, then the protocol used over reference point Ua shall be terminated.

Once the run of the protocol used over Ua reference point is completed the purpose of bootstrapping is fulfilled as it enabled the UE and NAF to use Ua reference point in a secure way.

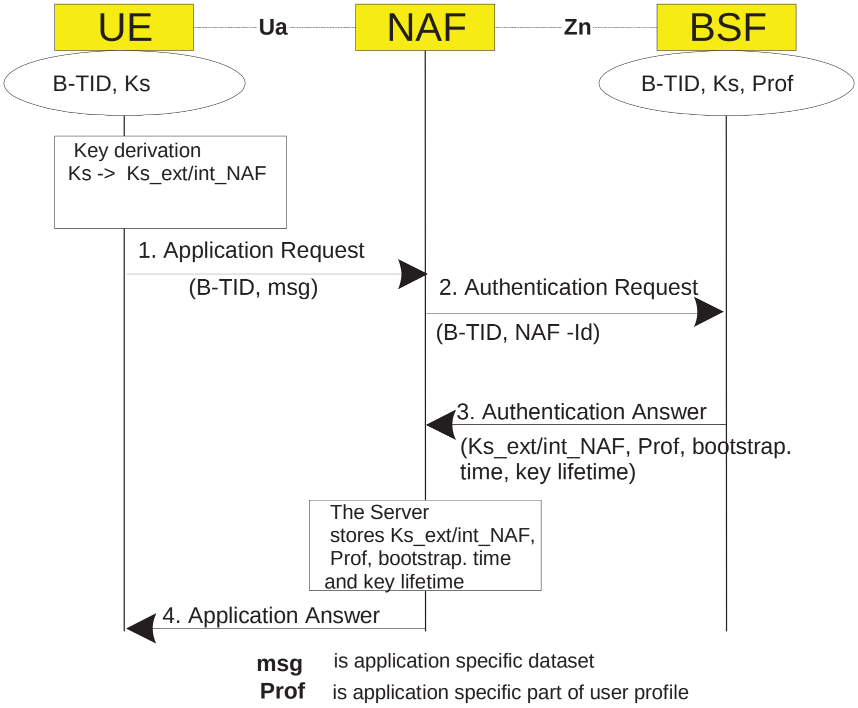

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation
Ks -> Ks_ext/int_NAF
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID, NAF -Id)
3. Authentication Answer
(Ks_ext/int_NAF, Prof, bootstrap. time, key lifetime)
The Server stores Ks_ext/int_NAF, Prof, bootstrap. time and key lifetime
4. Application Answer
msg is application specific dataset
Prof is application specific part of user profile

Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

The text from clause 4.5.4 of this document applies also here.

Annex A:
(Void)

Annex B (normative): Specification of the key derivation function KDF

B.1 Introduction

This annex specifies the key derivation function (KDF) that is used in the NAF specific key derivation in GBA (i.e. GBA_ME), GBA_U and GBA Digest. This KDF is also used to derive the TMPI. The key derivation function defined in the annex takes the following assumptions:

1. the input parameters to the key derivation functions are octet strings - not bit strings of arbitrary length:
2. a single input parameter will have lengths no greater than 65535 octets.

B.2 Generic key derivation function

The input parameters and their lengths shall be concatenated into a string S as follows:

1. The length of each input parameter measured in octets shall be encoded into a two octet-long string:

   a) express the number of octets in input parameter Pi as a number k in the range [0, 65535].

   b) Li is then a 16-bit long encoding of the number k, encoded as described in clause B.2.1.

2. String S shall be constructed from n+1 input parameters as follows:

   S = FC || P0 || L0 || P1 || L1 || P2 || L2 || P3 || L3 ||... || Pn || Ln

   where

   FC is single octet used to distinguish between different instances of the algorithm,

   P0 ... Pn are the n+1 input parameter encodings, and

   L0 ... Ln are the two-octet representations of the length of the corresponding input parameter encodings P0.. Pn.

   In this specification the following restriction applies to P0: P0 is a static ASCII-encoded string.

   This restriction is not part of the KDF definition and does not apply to the KDF when used by other 3GPP specifications unless explicitly stated so in those specifications.

3. The final output, i.e. the derived key is equal to the KDF computed on the string S using the key, denoted Key. The present document defines the following KDF:

   derived key = HMAC-SHA-256 (Key , S)

   as specified in [22] and [23].

B.2.1 Input parameter encoding

B.2.1.1 General

This clause specifies how encodings of different data types is to be done. Encoding rules for further data types may be added in future releases if needed.

B.2.1.2 Character string encoding

A character string shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [24] and apply Normalization Form KC (NFKC) as specified in [37].

B.2.1.3 Non-negative integer encoding

A non-negative integer value j, which is input to the KDF, shall be encoded into the octet string Pi as follows:

1. Encode j in base 2.

2. Let n be number of bits as defined for this input parameter in 3GPP specifications, or, if undefined elsewhere, let n be the least multiple of 8 that is greater or equal to the number of bits of the base 2 encoding of j

3. Create an octet string such that the least significant bit of the octet string shall be equal to the least significant bit of j written in base 2, the second to least significant bit of the octet string shall be equal to the second to least significant bit of j written in base 2 and so on (according to clause 3.4 of this specification). If the number of bits in j written in base 2 is less than n, the remaining most significant bits in the octet string shall be set to zero; if the number of bits in j written in base 2 is equal to n, there is no need for such zero padding.

The encoding Li of the number k of octets in Pi follows the above rule.

EXAMPLE1: If Pi is Uplink NAS COUNT then the number k of octets in Pi is 4, according to TS 33.401, Annex A, and n = 32. Assume that Uplink NAS COUNT has the integer value j = 259. Then the base 2 encoding of 259 is 100000011, and Pi consists of the octet string 0000000 0000000 00000001 00000011, or 0x00 0x00 0x01 0x03 in hex representation, and Li consists of the octet string 0000000 00000100, or 0x00 0x04 in hex representation.

EXAMPLE2: If the length of Pi is undefined elsewhere, and the integer value j = 259 is to be encoded into the parameter Pi then the base 2 encoding of 259 is 100000011, n and hence the length of parameter Pi in bits is 16, the number k of octets in Pi is 2, and Pi consists of the octet string 00000001 00000011, or 0x01 0x03 in hex representation, and Li consists of the octet string 0000000 00000010, or 0x00 0x02 in hex representation.

B.2.2 FC value allocations

FC values allocated for this specification shall be in range of 0x00 – 0x0F.

FC values allocated for TS 33.401 [35] shall be in range of 0x10 – 0x1F.

FC values allocated for TS 33.402 [36] shall be in range of 0x20 – 0x2F.

FC values allocated for TS 33.102 [2] shall be in range of 0x30 – 0x3F.

FC values allocated for TS 33.224 [39] shall be in range 0x40 – 0x48.

FC values allocated for TS 33.303 [49] shall be in range 0x49 – 0x4F.

FC values in range 0x50 – 0xFF are reserved for future use.

B.3 NAF specific key derivation in GBA and GBA_U

In GBA and GBA_U, the input parameters for the key derivation function shall be the following:

- FC = 0x01,

- P1 = RAND,

- L1 = length of RAND is 16 octets (i.e. 0x00 0x10),

- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1),

- L2 = length of IMPI is variable (not greater that 65535),
- P3 = NAF_ID with the FQDN part of the NAF_ID encoded to an octet string using UTF-8 encoding (see clause B.2.1), and
- L3 = length of NAF_ID is variable (not greater that 65535).

In the key derivation of Ks_NAF as specified in clause 4 and Ks_ext_NAF as specified in clause 5,

- P0 = "gba-me" (i.e. 0x67 0x62 0x61 0x2d 0x6d 0x65), and
- L0 = length of P0 is 6 octets (i.e., 0x00 0x06).

In the key derivation of Ks_int_NAF as specified in clause 5,

- P0 = "gba-u" (i.e. 0x67 0x62 0x61 0x2d 0x75), and
- L0 = length of P0 is 5 octets (i.e., 0x00 0x05).

The Key to be used in key derivation shall be:

- Ks (i.e. CK || IK concatenated) as specified in clauses 4 and 5,

NOTE: In the specification this function is denoted as:
Ks_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id),
Ks_ext_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id), and
Ks_int_NAF = KDF (Ks, "gba-u", RAND, IMPI, NAF_Id).

B.4 Derivation of TMPI

Derivation of TMPI follows the same procedure as NAF specific key derivation in GBA and GBA_U (see clause B.3). As the TMPI is stored in ME, for GBA_U the procedure for derivation of Ks_ext_NAF is followed.

NOTE: This procedure was chosen to avoid any changes to existing UICCs in case of GBA_U.

The BSF_Id defined in this clause consists of the full DNS name of the BSF as used for B-TID generation (see clause 4.5.2), concatenated with the Ua security protocol identifier for TMPI as specified in Annex H.

In GBA and GBA_U, the input parameters for the key derivation function to derive the TMPI shall be the following:

- FC = 0x01,
- P0 = "gba-me" (i.e. 0x67 0x62 0x61 0x2d 0x6d 0x65) (see clause B.3 for Ks_NAF and Ks_ext_NAF),
- L0 = length of P0 is 6 octets (i.e., 0x00 0x06).
- P1 = RAND,
- L1 = length of RAND is 16 octets (i.e. 0x00 0x10),
- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1),
- L2 = length of IMPI is variable (not greater that 65535),
- P3 = the BSF_Id encoded to an octet string using UTF-8 encoding (see clause B.2.1), and
- L3 = length of P3 is variable (not greater that 65535).

The Key to be used in derivation of TMPI shall be:

- Ks (i.e. CK || IK concatenated) as specified in clauses 4 and 5.

The TMPI is then computed as TEMP@tmpi.bsf.3gppnetwork.org where TEMP is the base64-encoding [12] of the 24 most significant octets of the output of KDF.

B.5 Derivation of passwd and Ks

Derivation of passwd and Ks for GBA_Digest shall follow the same procedure as NAF specific key derivation in GBA and GBA_U as specified in clause B.3.

The input parameters for the key derivation function to derive passwd and Ks shall be the following:

- FC = 0x01,
- P1 = TLS_MK_Extr,
- L1 = length of TLS_MK_Extr is 48 octets (i.e. 0x00 0x30),

In the derivation of passwd as specified in clause M.6.3, step 5,

- P0 = "GBA_Digest_RESP"
  (i.e. 0x47 0x42 0x41 0x5F 0x44 0x69 0x67 0x65 0x73 0x74 0x5F 0x52 0x45 0x53 0x50), and
- L0 = length of P0 is 15 octets (i.e., 0x00 0x0F).

In the key derivation of Ks as specified in clause M.6.3, step 6,

- P0 = "GBA_Digest_Ks"
  (i.e. 0x47 0x42 0x41 0x5F 0x44 0x69 0x67 0x65 0x73 0x74 0x5F 0x4B 0x73),
- L0 = length of P0 is 13 octets (i.e., 0x00 0x0D),
- P2 = RESP, and
- L2 = length of RESP is variable and depends on the algorithm used in HTTP Digest (e.g., 32 if MD5 is used).

The Key to be used in key derivation function shall be:

- H(A1) as specified in clause M.6.3, step 5.

NOTE: In the present document this function is denoted as:
passwd = KDF (H(A1), "GBA_Digest_RESP", TLS_MK_Extr), and
Ks = KDF (H(A1), " GBA_Digest_Ks", TLS_MK_Extr, RESP).

B.6 NAF specific key derivation in GBA_Digest

In GBA_Digest, the input parameters for the key derivation function to derive Ks_NAF shall be the following:

- FC = 0x01;
- P0 = "gba-digest" (i.e. 0x67 0x62 0x61 0x2d 0x64 0x69 0x67 0x65 0x73 0x74);
- L0 = length of P0 is 10 octets (i.e., 0x00 0x0a);
- P1 = nonce;
- L1 = length of nonce is variable (not greater than 65535);
- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1 of the present document);
- L2 = length of IMPI is variable (not greater than 65535);
- P3 = NAF_ID with the FQDN part of the NAF_ID encoded to an octet string using UTF-8 encoding (see clause B.2.1 of the present document;
- L3 = length of NAF_ID is variable (not greater that 65535).

The Key to be used in key derivation shall be:

- Ks as specified in clause B.5 of the present document.

NOTE: In clause M.6.3 this function is denoted as:
Ks_NAF = KDF (Ks, "gba-digest", nonce, IMPI, NAF_Id).

Annex C:
(Void)

Annex D (informative): Dialog example for user selection of UICC application used in GBA

For certain cases, clause 4.4.8 specifies user involvement in the selection of the UICC application used for GBA procedures. A dialog window example for such an involvement is described below:

- The title of the dialog: "Authentication request".
- Explanation: "A service requires you to authenticate, please select your identity:"
- List of identities: A selectable list of applications on the UICC. The text visible for each application is extracted from the "Label" field of the application list on the UICC.
- Buttons: "Select" and "Cancel".

Annex E (normative): TLS profile for securing Zn/Zn' reference points

This Annex applies for the Zn' reference point when using DIAMETER or HTTP, and applies for the Zn reference point if using HTTP.

The TLS profile is specified in TS 33.310 [19], Annex E and shall apply with the addition that TLS 1.0 [28] shall also be supported. For all TLS versions the provisions on ciphersuites given in TS 33.310 [19], Annex E, shall apply. The TLS endpoints shall mutually authenticate using certificates as part of TLS session establishment.

NOTE: It is likely that support of TLS 1.0 will no longer be mandatory in a future 3GPP release.

The TLS certificates shall follow the requirements in clause 6.1 of TS 33.310 [19] for TLS certificates, with the exceptions as given in the following.

The Zn-Proxy certificate, i.e. the client certificate used in TLS handshake, shall contain the subjectAltName extension with one or more dNSName names. The dNSName name may contain the wildcard character '*' and the matching is performed as specified in RFC 2818 [18] section 3.1.

The Zn-Proxy certificate shall contain all the DNS names of NAFs that may send a request for NAF specific shared secret through the Zn-Proxy to the subscriber's home BSF. If a new NAF is added, the new DNS name is either covered in the certificate by using the wildcard character approach (e.g. "*.operator.com"), or a new dNSName name needs to be added to the certificate. In the latter case, new certificate is needed for the Zn-Proxy.

Annex F (informative): Handling of TLS certificates

An authentication framework for TLS is available [19].

Annex G (normative): GBA_U UICC-ME interface

This annex describes the UICC-ME interface to be used when a GBA_U aware UICC application is active and the ME is involved in a GBA bootstrapping procedure. When the UICC application is not GBA_U aware, the ME uses AUTHENTICATE command in non-GBA_U security context (i.e. UMTS security context in case of USIM application and IMS security context in case of the ISIM) as defined in TS 31.102 [1] and TS 31.103 [10].

G.1 GBA_U Bootstrapping procedure

This procedure is part of the Bootstrapping procedure as described in clause 5.3.2.

The ME sends RAND and AUTN to the UICC, which performs the Ks derivation as described in clause 5.3.2.

The UICC then stores Ks. The UICC also stores the used RAND to identify the current bootstrapped values. RAND value in the UICC shall be further accessible by the ME.

The ME then finalizes the Bootstrapping procedure and stores in the UICC the Transaction Identifier (B-TID) and Key Life Time associated with the previous bootstrapped keys (i.e. Ks). Transaction Identifier and Key Life Time values in the UICC shall be further accessible by the ME.

At the end of the GBA_U bootstrapping procedure the UICC stores Ks, Transaction Identifier, Key Life Time and the RAND.

The UICC sends RES to the ME.

A new bootstrapping procedure replaces Ks, B-TID, Key LifeTime and RAND values of the previous bootstrapping procedure.

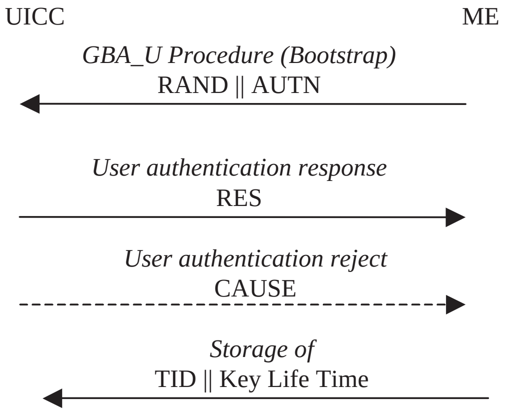

UICC
ME
GBA_U Procedure (Bootstrap)
RAND || AUTN
User authentication response
RES
User authentication reject
CAUSE
Storage of
TID || Key Life Time

Figure G.1: GBA_U Bootstrap Procedure

G.2 GBA_U NAF Derivation procedure

This procedure is part of the Procedures using bootstrapped Security Association as described in clause 5.3.3

The ME sends NAF_ID and IMPI to the UICC. The UICC then performs Ks_ext_NAF and Ks_int_NAF derivation as described in clause 5.3.2. The UICC uses the RAND and Ks values stored from the previous bootstrapping procedure. The UICC returns Ks_ext_NAF to the ME and stores Ks_int_NAF and associated B-TID together with NAF_Id.

In case that the UICC does not have enough storage available for the generated Ks_int_NAF and associated parameters, the UICC shall overwrite an existing Ks_int_NAF entry (Ks_int_NAF and associated parameters). To determine the Ks_int_NAF to overwrite, the UICC shall construct a list of Ks_int_NAF entry numbers by storing in the list first position the entry number of the last used or derived Ks_int_NAF and by shifting down the remaining list elements. The last Ks_int_NAF entry number in this list corresponds to the Ks_int_NAF to overwrite when the UICC runs out of free records.

If an existing Ks_int_NAF entry in use is overwritten, the application Ks_int_NAF shall not be affected (e.g. in case a Ks_int_NAF was put into use as an MBMS MUK key, the MUK key shall continue to be available for the MBMS application).

NOTE: A previous GBA_U Bootstrap needs to be undertaken before. If Ks is not available in the UICC, the command will answer with the appropriate error message.

The input parameters IMPI and the FQDN part of NAF_ID shall be encoded as specified Annex B.2.1.

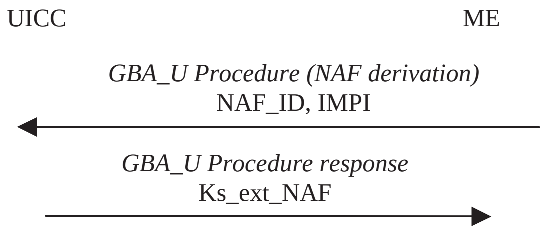

UICC
ME
GBA_U Procedure (NAF derivation)
NAF_ID, IMPI
GBA_U Procedure response
Ks_ext_NAF

Figure G.2: GBA_U NAF derivation procedure

Annex H (normative): Ua security protocol identifier

H.1 Definition

The Ua security protocol identifier is a string of five octets. The first octet denotes the organization which specifies the Ua security protocol. The four remaining octets denote a specific security protocol within the responsibility of the organization.

For all Ua protocols specified by 3GPP this Annex shall contain a complete list of these protocols. For Ua protocols specified by other organizations this Annex shall only specify the organization octet of the Ua security protocol identifier. Two organization octets are reserved for special use.

H.2 Organization Octet

The organization octet denotes the organization specifying the particular protocol. Each organization intending to specify a Ua security protocol identifier shall apply to 3GPP to receive an organization octet value, which shall be registered within this Annex. Following is a list of registered organization octets:

"0x00" as first octet is the default value for protocols not specified otherwise. When octet "0x00" is used as first octet, only Ua security protocol identifier (0x00,0x00,0x00,0x00,0x00) shall be used.

NOTE 1: All protocols having this Ua security protocol identifier cannot be separated from each other.

"0x01" .. "0xFE" as the first octet denote organizations specifying Ua security protocol identifiers.

"0xFF" as the first octet denotes the private range of Ua security protocol identifiers.

NOTE 2: identifiers with "0xFF" as first octet may be used for defining local/experimental protocols without need for registration. When using such an identifier, however, it may happen that a security breach in one security protocol over Ua can be exploited by an attacker to mount successful attacks on a different security protocol over Ua.

The following values for organizations are assigned:

"0x01"3GPP

NOTE 3: All protocols having the organization octet "0x01" are specified in annex H.3.

"0x02"3GPP2

"0x03"Open Mobile Alliance

"0x04"GSMA

H.3 Ua security protocol identifiers for 3GPP specified protocols

The following Ua security protocol identifiers are specified by 3GPP:

(0x01,0x00,0x00,0x00,0x00)	Ua security protocol according to TS 33.221 [5].
(0x01,0x00,0x00,0x00,0x01)	Ua security protocols according to TS 33.246 [26].

NOTE 1: TS 33.246 [26] provides key separation between the keys that are used within HTTP digest and MIKEY protocols.

(0x01,0x00,0x00,0x00,0x02) Ua security protocol HTTP digest authentication according to TS 24.109 [29], unless HTTP digest authentication is used in the context of another Ua security protocol, which is already covered elsewhere in this Annex.

(0x01,0x00,0x00,0x00,0x03) Ua security protocols used with HTTP-based security procedures for MBMS user services according to TS 26.237 [38].

(0x01,0x00,0x00,0x00,0x04) Ua security protocols used with SIP-based security procedures for MBMS user services according to TS 26.237 [38].

(0x01,0x00,0x00,0x00,0x05) Ua security protocols used with Generic Push Layer according to TS 33.224 [39], unless Generic Push Layer is used in the context of another Ua security protocol, which is already covered elsewhere in this Annex.

(0x01,0x00,0x00,0x00,0x06) Ua security protocol for IMS UE to KMS http based message exchanges according to "IMS media plane security", TS 33.328 [40]

(0x01,0x00,0x00, 0x01,0x00) Generation of TMPI according to Annex B.4.

NOTE 2: This protocol identifier is not strictly a Ua protocol identifier, but its use in key derivation function is exactly equal.to a Ua protocol identifier.

(0x01,0x00,0x01,yy,zz) Ua security protocol for "Shared key-based UE authentication with certificate-based NAF authentication", according to TS 33.222 [25] section 5.3, or "Shared key-based mutual authentication between UE and NAF", according to TS 33.222 [25] section 5.4. Here, "yy,zz" is the protection mechanism CipherSuite code according to the defined values for TLS CipherSuites in the IANA TLS Cipher Suite Registry which is referenced in TLS V1.2 [41].

NOTE 3: The "Certificate based mutual authentication between UE and NAF' according to TS 33.222 [25] section 5.5 does not require a Ua protocol identifier.

NOTE 4: As an example: RFC 5246 [41] CipherSuite TLS_RSA_WITH_3DES_EDE_CBC_SHA has code = { 0x00,0x0A }, thus the according protocol identifier shall be (0x01,0x00,0x01,0x00,0x0A).

(0x01,0x00,0x02,yy,zz) Ua security protocol for "Shared key-based UE authentication with certificate-based NAF authentication", according to TS 33.222 [25] Annex D. Here, "yy,zz" is the protection mechanism CipherSuite code according to the defined values for TLS CipherSuites in the IANA TLS Cipher Suite Registry which is referenced in TLS V1.2 [41]. This Ua security protocol identifier is used for the case outlined in TS 33.222 [5] Annex D, where e.g. HTML FORM based authentication is used within a TLS tunnel.

NOTE 4: The third octet (0x02) distinguish this case from other protocols tunneled inside the TLS tunnel.

Annex I (normative): 2G GBA

I.0 Introduction

This annex specifies the implementation option to allow the use of SIM cards or SIMs on UICC for GBA. The procedure specified in this annex is called 2G GBA. 2G GBA allows access to applications in a more secure way than would be possible with the use of passwords or with GSM without enhancements. It may be useful for operators who have not yet fully deployed USIMs.

The usage of the term 2G GBA in this specification does not restrict the usage of GBA over only 2G access networks i.e. GSM access. Similarly the use of the term 3G GBA in this specification does not restrict the usage of GBA over only 3G access networks i.e. UMTS. In this specification the term 2G GBA refers to the usage of a SIM card or SIM on UICC, while 3G GBA or GBA on its own, refers and to the usage of a USIM/ISIM on a UICC.

Clauses 4 and 5 of the present document do not apply to this annex unless explicitly stated.

I.1 Reference model

The reference model is the same as described in section 4.1.

I.2 Network elements

I.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the 2G AKA protocol and the TLS protocol, and agree on session keys that are afterwards applied between UE and a Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using the key derivation procedure as specified in Annex B. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause I.5.2.

The BSF shall be able to acquire the GBA user security settings (GUSS) from the HSS.

The BSF shall be able to discover from the type of authentication vectors sent by the HSS whether the subscriber is a 2G or a 3G subscriber.

The BSF shall be able to keep a list, which assigns NAFs to NAF Groups. This list is used to select if any and which application-specific USS within GUSS is valid for a certain NAF.

NOTE 1: The operator does the assignment of NAFs to NAF Groups. NAF Group definitions in HSS and all connected BSFs belonging to the same operator's network shall be equal (cf., clause I.2.3). As these network elements belong to the same operator's network, standardisation of the NAF Group definitions themselves is not necessary in 3GPP.

NOTE 2: The NAF grouping may be e.g. "home" and "visited". It allows the BSF to send USSs for the same application with e.g. different authorization flags to different NAFs, e.g., in home network and visited networks. The NAF e.g. in visited network indicates only the requested application, but it is unaware of the grouping in home network of the subscriber.

NOTE 3: If support of GBA User Security Settings (GUSS) for service differentiation is desired in combination with HLR, then this can be achieved by addition of a database to the BSF containing the needed GUSS information.

The BSF shall allow the operator to configure a BSF policy whether to accept 2G subscribers or not for a certain NAF.

I.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and a NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of a NAF are:

- there need not be a previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire zero or more application-specific USSs from the HSS via the BSF;
- NAF shall be able to set the local validity condition of the shared key material according to the local policy;
- NAF shall be able to check lifetime and local validity condition of the shared key material;
- NAF shall have a policy whether to accept 2G subscribers. However, whether the SIM card is allowed to be used with a specific Ua application or not, is dependent on the relevant Ua application. If there is a specific TS for an application using a particular Ua protocol, and unless this TS explicitly prohibits the use of SIM, the operator is allowed to configure a NAF policy whether to accept 2G subscribers or not for this Ua application.

NOTE: Without additional measures, GBA does not guarantee the freshness of the key, Ks(_int/ext)_NAF in the sense that it does not guarantee that the key was not used in a previous run of the Ua protocol. The additional measures which may be taken by the UE and the NAF to ensure key freshness in GBA are:

1) enforce a new run of the Ub protocol (thus generating a new Ks) before deriving a new Ks_NAF.

2) store previously used keys Ks(_int/ext)_NAF, or the corresponding key identifiers B-TID, until the end of their lifetime.

A UE and a NAF that support a Ua protocol that does not provide replay protection over unconnected runs of the protocol, will need to take corresponding action to avoid replay attacks if desired.

I.2.2a Zn-Proxy

The text from section 4.2.2a applies also here.

I.2.3 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS.

The requirements on the HSS are:

- HSS shall provide the only persistent storage for GUSSs;
- GUSS shall be defined in such a way that interworking of different operators for standardised application profiles is possible;
- GUSS shall be defined in such a way that profiles for operator specific applications and extensions to existing application profiles are supported without need for standardisation of these elements.
- GUSS shall be able to contain application-specific USSs that contain parameters that are related to identification or authorization information of one or more applications hosted by one ore more NAFs. Any other types of parameters are not allowed in the application-specific USS.

NOTE 1: The necessary subscriber profile data may be fetched by the NAF from its local database.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - subscriber specific key lifetime;
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or subscriber does not have GUSS then the BSF will use the default values in the BSF local policy defined by the particular MNO.

- HSS shall be able to assign application-specific USSs to a NAF Group. This shall be defined in such a way that different USSs for the same application, but for different groups of NAFs, are possible. The restrictions on the number of USSs per GUSS are dependent on the usage of NAF Groups by the operator:
  - if no NAF Groups are defined for this application then at most one USS per application is stored in GUSS;
  - if NAF Groups are defined for this application then at most one USS per application and NAF Group is stored in GUSS.
- NAF Group definitions in the HSS and all connected BSFs belonging to the same operator's network shall be equal.
  - Information on UICC type and on key choice are not required for 2G subscribers. 2G GBA is regarded as ME-based.

I.2.4 UE

The required functionalities from the UE are:

- the support of HTTP Digest AKA protocol;
- the support of TLS;
- the capability to use a SIM in bootstrapping;
- the capability for a Ua application on the ME to indicate to the GBA Function on the ME whether a SIM is allowed for use in bootstrapping (see clause I.4.8);
- the capability to derive new key material to be used with the protocol over Ua interface from Kc, RAND, SRES and Ks-input;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]).

A 2G GBA-aware ME shall support both 3G GBA_U, as specified in clause 5.2 and 3G GBA_ME procedures, as specified in clause 4.5.

The security of2G GBA relies on that the ME hasimplemented the following properties:

- The ME sets each fill bit it sends to a random value, in accordance with clause 5.2 of TS 44.006 [46].

NOTE: This requirement is fulfilled by MEs from Rel-8 onwards.

- The ME does not implement GEA1, in accordance with clause D.4.9 of TS 43.020 [47].

NOTE: This requirement is fulfilled by MEs from Rel-12 onwards.

I.2.5 SLF

The text from section 4.2.5 applies also here.

I.2.6 HLR

The requirement on the HLR is the same as in clause 4.3.6.

I.3 Bootstrapping architecture and reference points

I.3.1 Reference point Ub

The reference point Ub is between the UE and the BSF. Reference point Ub provides mutual authentication between the UE and the BSF. It allows the UE to bootstrap the session keys based on 2G AKA infrastructure.

I.3.2 Reference point Ua

The reference point Ua carries the application protocol, which is secured using the keys material agreed between UE and BSF as a result of the run of the protocol over reference point Ub.

I.3.3 Reference point Zh

The reference point Zh used between the BSF and the HSS allows the BSF to fetch the required authentication information and all GBA user security settings from the HSS. The reference point Zh is an intra-operator domain interface. The interface to the 2G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

I.3.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous protocol run over the reference point Ub from the UE to the BSF. It is also used to fetch application-specific user security settings from the BSF, if requested by the NAF.

I.3.5 Reference point Dz

The text from section 4.3.5 applies also here.

I.3.6 Reference point Zh'

The optional reference point Zh' used between the BSF and the HLR allows the BSF to fetch the required authentication information. The reference point Zh' is an intra-operator domain interface.

I.4 Requirements and principles for bootstrapping

I.4.0 General requirements

The following requirements and principles are applicable to bootstrapping procedure:

- the 2G GBA bootstrapping function shall not depend on the particular NAF;
- the server implementing the bootstrapping function needs to be trusted by the home operator to handle authentication vectors;
- the server implementing the NAF needs only to be trusted by the home operator to handle derived key material;

- it shall be possible to support NAF in the operator's home network and in the visited network;
- the architecture shall not preclude the support of network application function in a third network;
- to the extent possible, existing protocols and infrastructure should be reused;
- in order to ensure wide applicability, all involved protocols are preferred to run over IP;
- it shall be prevented that a security breach in one NAF who is using the GBA, can be used by an attacker to mount successful attacks to the other NAFs using the GBA.
- an attacker shall not be able to exploit a security breach in one security protocol over Ua in order to mount a successful attack against a different security protocol over Ua.
- Existing SIM cards or SIMs on UICCs and their specifications shall not be impacted.
- If USIM or ISIM are available they shall be used as specified in sections 4 and 5, and 2G GBA shall not be used.
- 2G GBA shall not impact the USIM / ISIM based GBA as specified in sections 4 and 5.
- 2G GBA shall not reduce security for USIM / ISIM users.
- 2G GBA shall minimise the changes to the USIM / ISIM based GBA specified in section 4.
- 2G GBA shall provide measures to mitigate known vulnerabilities of GSM.

I.4.1 Access Independence

Bootstrapping procedure is access independent. Bootstrapping procedure requires IP connectivity from UE.

I.4.2 Authentication methods

Authentication between the UE and the BSF shall not be possible without a valid cellular subscription. Authentication shall be based on the GSM authentication (also called 2G AKA) protocol. BSF authentication shall in addition be based on TLS with server certificates.

I.4.3 Roaming

The text from section 4.4.3 applies also here.

I.4.4 Requirements on reference point Ub

The requirements for reference point Ub are:

- the BSF shall be able to identify the UE;
- the BSF and the UE shall be able to authenticate each other based on the methods in I.4.2;
- the BSF shall be able to send a bootstrapping transaction identifier to the UE;
- the UE and the BSF shall establish shared keys;
- the BSF shall be able to indicate to the UE the lifetime of the key material. The key lifetime sent by the BSF over Ub shall indicate the expiry time of the key.

NOTE: This does not preclude a UE to refresh the key before the expiry time according to the UE's local policy.

I.4.5 Requirements on reference point Zh

The requirements for reference point Zh are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: This requirement may be fulfilled by physical or proprietary security measures since BSF and HSS are located within the same operator's network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;
- optionally the BSF may have the capability able to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);
- the HSS shall be able to send one 2G AKA vector at a time to the BSF;
- the HSS shall be able to send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

NOTE 2: If subscriber's GUSS is updated in HSS, this is not propagated to the BSF. The GUSS in the BSF is updated when the BSF next time fetches the authentication vectors and GUSS from the HSS over Zh reference point as part of the bootstrapping procedure.

- no state information concerning bootstrapping shall be required in the HSS;
- all procedures over reference point Zh shall be initiated by the BSF;
- the number of different interfaces to HSS should be minimized.

I.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;
- If the BSF and the NAF are located within the same operator's network, the DIAMETER based Zn reference point shall be secured according to NDS/IP [13];
- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E of the present document;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E of the present document;
- The BSF shall verify that the requesting NAF is authorised to obtain the key material or the key material and the requested USS;
- The NAF shall be able to send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall be able to verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The BSF shall be able to send the requested key material to the NAF;
- The NAF shall be able to get a selected set of application-specific USSs from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall be able to indicate to the BSF the single application or several applications it requires USSs for;

NOTE 2: If some application needs only a subset of an application-specific USS the NAF selects this subset from the complete set of USS sent from BSF.

- The BSF shall be able to be configured on a per NAF or per application basis if private subscriber identity and which application-specific USSs may be sent to a NAF;

NOTE 3: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI (but then there is no user anonymity). If the BSF does not send the IMPI or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- If a NAF requests USSs from the BSF and they are not present in subscriber's GUSS, it shall not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF shall then send only the requested and found USSs to the NAF;

- It shall be possible to configure a local policy as follows: BSF may require one or more application-specific USS to be present in a particular subscriber's GUSS for a particular requesting NAF, and to reject the request from the NAF in case the conditions are not fulfilled. In order to satisfy this local policy, it is not required that the NAF requests the USSs over the Zn reference point, which the BSF requires to be present in the GUSS, rather it is sufficient that the BSF checks the presence of the USSs locally. It shall also be possible configure the BSF in such a way that no USS is required for the requesting NAF;

- The BSF shall be able to indicate to the NAF the bootstrapping time and the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 4: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 5: If one or more of the USSs that have been delivered to the NAF has been updated in subscriber's GUSS in the HSS, this change is propagated to the NAF the next time it fetches the USS from the BSF over Zn reference point (provided that the BSF has updated subscriber's GUSS from the HSS over Zh reference point).

- The BSF shall send information to the NAF that the subscriber is a 2G subscriber. If no such information is sent the NAF shall assume that the subscriber is a 3G subscriber.

NOTE 6: This requirement enables a NAF to accept 2G subscribers according to its local policy. The second sentence ensures backward compatibility with the procedures specified in section 4 and 5 of this specification. Note also that inclusion of information on the type of subscription in the GUSS would not suffice to satisfy this requirement as a GUSS need not be present for every subscriber.

- The BSF may determine according to its local policy that the NAF shall not serve 2G subscribers. If this is the case, the BSF does not send keys to the NAF.

NOTE 7: This requirement allows an operator controlling the BSF to determine which applications shall use 3G security only. This requirement is also necessary for NAFs, which are not capable to evaluate the information about the subscription type sent by the BSF, e.g. pre-release 7 NAFs.

- NAF shall be able to indicate to BSF the protocol identifier of Ua security protocol it requires the key material by sending NAF-Id to BSF (cf. Annex H).

I.4.7 Requirements on Bootstrapping Transaction Identifier

Bootstrapping transaction identifier (B-TID) shall be used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

Requirements for B-TID are:

- B-TID shall be globally unique;

- B-TID shall be usable as a key identifier in protocols used in the reference point Ua;

- NAF shall be able to detect the home network and the BSF of the UE from the B-TID.

NOTE 1: NAF can remove the security association based on deletion conditions after the key has become invalid.

NOTE 2: Care has to be taken that the parallel use of GBA and non-GBA authentication between UE and NAF does not lead to conflicts, e.g. in the name space. This potential conflict cannot be resolved in a generic way as it is dependent on specific protocol and authentication mechanism used between UE and application server. It is therefore out of scope of this specification.
For the example of HTTP Digest authentication used between UE and NAF, parallel use is possible as the following applies: <username,password>-pairs must be unique to one realm only. As the NAF controls the realm names, it has to ensure that only the GBA based realm is named with the reserved 3GPP realm name. In the special case that the NAF wants to allow non GBA based authentication in the GBA realm also, it has to ensure that no usernames in the format of a B-TID are used outside GBA based authentication.

I.4.8 Requirements on selection of UICC application and SIM card

If a UICC is present in the UE, containing a USIM or an ISIM, then a USIM or ISIM shall be used as specified in section 4.4.8. Otherwise a SIM shall be used.

If no UICC, but a SIM card is present in the UE, the SIM card shall be used. The IMPI is obtained from the IMSI as specified in section 4.4.8.

I.4.9 Requirements on reference point Ua

The text from section 4.4.9 applies also here.

I.4.10 Requirements on reference point Dz

The text from section 4.4.10 applies also here.

I.4.11 Requirements on reference point Zh'

The requirements for reference point Zh' are the same as in clause 4.4.12.

I.5 Procedures

This chapter specifies in detail the format of the 2G GBA bootstrapping procedure that is further utilized by various applications. It contains the authentication procedure with BSF, and the key material generation procedure.

I.5.1 Initiation of bootstrapping

The text from clause 4.5.1 of the present document applies also here.

I.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure I.3). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. subclause I.5.3).

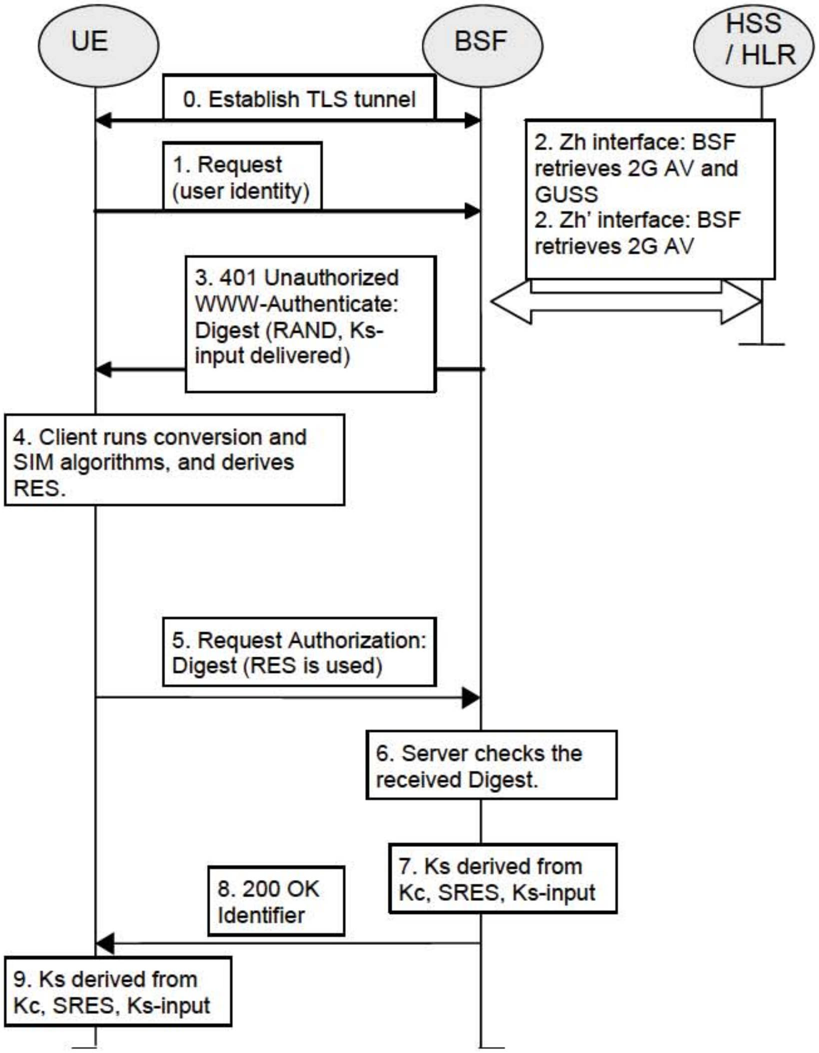

UE
BSF
HSS / HLR
0. Establish TLS tunnel
1. Request (user identity)
2. Zh interface: BSF retrieves 2G AV and GUSS
2. Zh' interface: BSF retrieves 2G AV
3. 401 Unauthorized WWW-Authenticate: Digest (RAND, Ks-input delivered)
4. Client runs conversion and SIM algorithms, and derives RES.
5. Request Authorization: Digest (RES is used)
6. Server checks the received Digest.
7. Ks derived from Kc, SRES, Ks-input
8. 200 OK Identifier
9. Ks derived from Kc, SRES, Ks-input

Figure I.3: The bootstrapping procedure

1. The UE sets up a confidentiality-protected TLS tunnel with the BSF. In the set up of the TLS tunnel, the UE shall authenticate the BSF by means of a certificate provided by the BSF. All further communication between ME and BSF is sent through this TLS tunnel. The UE now sends an initial HTTPS request.

2. The BSF requests authentication vectors and GUSS from the HSS over Zh. The HSS returns the complete set of GBA user security settings (GUSS) and one 2G authentication vectors (AV = RAND, SRES, Kc) over the Zh reference point. The BSF discovers that the UE is equipped with 2G SIM by looking at the type of authentication vectors.

   If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

   In the case that no HSS with Zh reference point support is deployed, the BSF requests the authentication vector from either an HSS with Zh' reference point support or an HLR over the Zh' reference point. The HLR or HSS with Zh' reference point support returns one 2G authentication vectors (AV = RAND, SRES, Kc) over the Zh' reference point. The BSF discovers that the UE is equipped with 2G SIM by looking at the type of authentication vectors.

The BSF converts one 2G authentication vector (RAND, Kc, SRES) to the parameter RES.

RES = KDF (key, "3gpp-gba-res", SRES), truncated to 128 bits

where key = Kc || Kc || RAND and KDF is the key derivation function specified in Annex B of TS 33.220.

The BSF shall also select a 128-bit random number "Ks-input" and set

server specific data = Ks-input
in the aka-nonce of HTTP Digest AKA, cf. [4].

NOTE 1: "Truncated to 128 bits" means that from the 256 bits output of KDF, the 128 bits numbered as [0] to [127] are used.

NOTE 2: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 2.

3. The BSF shall forward RAND and server specific data in the 401 message to the UE (without RES). This is to demand the UE to authenticate itself.

4. The UE extracts RAND from the message and calculates the corresponding Kc and SRES values. It then calculates the parameter RES from these values as specified in step 2.

5. The UE sends another HTTP request, containing the Digest AKA response (calculated using RES as the password) and a cnonce (cf. [3]), to the BSF.

6. The BSF authenticates the UE by verifying the Digest AKA response. If the authentication fails the BSF shall not re-use the authentication vector in any further communication.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

7. The BSF shall generate key material Ks by computing Ks = KDF (key, Ks-input, "3gpp-gba-ks", SRES).
The B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e. base64encoded(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

8. The BSF shall send a 200 OK message, including a B-TID and an authentication-info header (cf. [3]), to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks.

9. The UE shall abort the procedure if the server authentication according to [3] fails. If it is successful the UE shall generate the key material Ks in the same way as the BSF.

10. Both the UE and the BSF shall use the Ks to derive the key material Ks_NAF for use with the procedures specified in clause I.5.3. Ks_NAF shall be used for securing the reference point Ua.

Ks_NAF is computed as Ks_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_Id), where KDF is the key derivation function as specified in Annex B, and the key derivation parameters consist of the user's IMPI, the NAF_Id and RAND. The NAF_Id is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is specified in Annex H. KDF shall be implemented in the ME.

NOTE 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

To allow consistent key derivation based on NAF name in UE and BSF, at least one of the three following prerequisites shall be fulfilled:

(1)The NAF is known in DNS under one domain name (FQDN) only, i.e. no two different domain names point to the IP address of the NAF. This has to be achieved by administrative means.

(2) Each DNS entry of the NAF points to a different IP address. The NAF responds to all these IP addresses. Each IP address is tied to the corresponding FQDN by NAF configuration. The NAF can see from the IP address, which FQDN to use for key derivation.

(3) Ua uses a protocol which transfers the host name (FQDN of NAF as used by UE) to NAF (e.g. HTTP/1.1 with mandatory Host request header field). This requires the NAF to check the validity of the host name, to use this name in all communication with UE where appropriate, and to transfer this name to BSF to allow for correct derivation of Ks_NAF.
In case of a TLS tunnel over Ua this requires either multiple-identities certificates for the NAF or the deployment of RFC 3546 [9] over Ua or other protocol means with similar purpose over Ua.

The UE and the BSF shall store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated or until the deletion conditions are satisfied (see 4.4.11).

I.5.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause I.5.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with an NAF the following steps are executed as depicted in figure I.4.

1. UE starts communication over reference point Ua with the NAF:

   - in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:

     - if a key Ks for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause I.5.2;

     - if no key Ks for the selected UICC application is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

     If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF;

   - if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g. because the key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure I.5. If the key's lifetime has expired the protocol used over reference point Ua shall be terminated. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause I.5.2, in order to obtain a new key Ks.

     To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause I.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of clause I.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

   NOTE 1: If the shared key between UE and NAF is invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

   - the UE supplies the B-TID to the NAF, in the form as specified in clause I.3.2, to allow the NAF to retrieve the corresponding keys from the BSF;

   NOTE 2: The UE may adapt the key material Ks_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification.

- the key management procedures for GBA related keys in the ME (i.e. Ks and Ks_NAF keys) are described in section 4.4.11.

- when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

  According to the procedures defined in clauses I.5.2 and I.5.3, in the UE there is at most one Ks_NAF key stored per NAF-Id.

2. NAF starts communication over reference point Zn with BSF

   - The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua.;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF-Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

3. The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause I.5.2, and supplies to NAF the requested key Ks_NAF, as well as the bootstrapping time and the lifetime of that key, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. In addition, the BSF shall indicate to the NAF that the subscriber is a 2G subscriber. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

   NOTE 4: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

   NOTE 5: The NAF will adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

   - The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause I.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.

   - The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy;

   - If the BSF or the NAF determined, according to their local policies, that the NAF shall not serve 2G subscribers, the NAF shall terminate the protocol over the reference point Ua.

   - When the NAF receives the Zn response, it shall check that the GBA type in the Zn response corresponds with the GBA type negotiated over Ua protocol. If this is not the case, NAF shall terminate the protocol over the reference point Ua.

4. NAF continues with the protocol used over the reference point Ua with the UE.

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way.

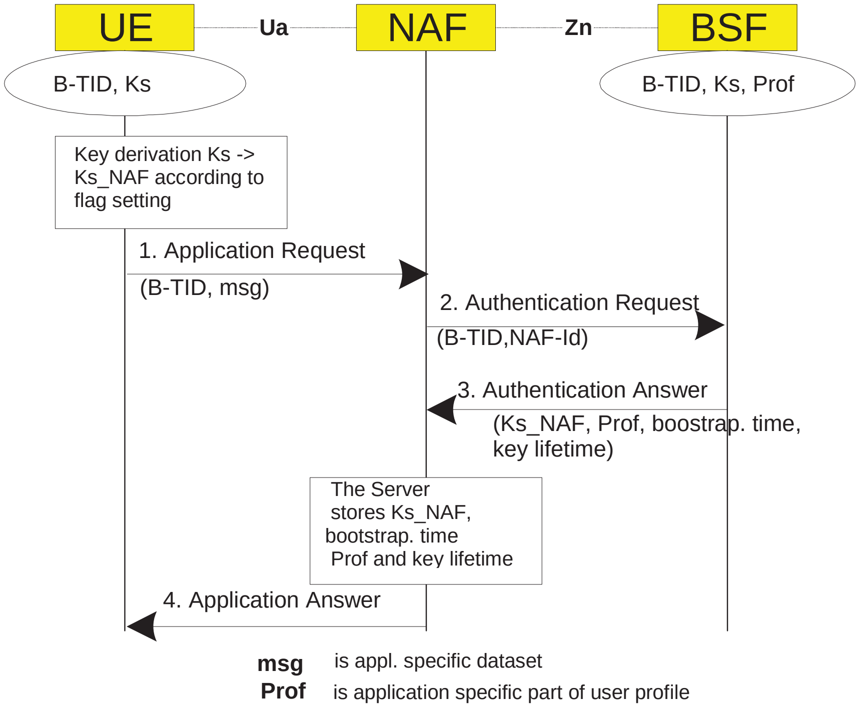

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID,NAF-Id)
3. Authentication Answer
(Ks_NAF, Prof, boostrap. time, key lifetime)
The Server stores Ks_NAF, bootstrap. time Prof and key lifetime
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure I.4: The bootstrapping usage procedure

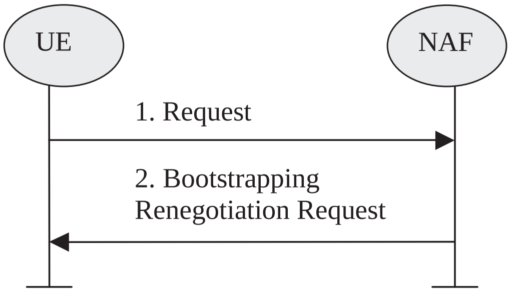


Figure I.5: Bootstrapping renegotiation request

I.5.4 Procedure related to service discovery

The UE shall discover the address of the BSF from the IMSI on the SIM. The same discovery procedure as specified in Section 4.5.4 shall be used.

I.6 TLS Profile

The UE and the BSF shall support TLS according to the TLS profile given in TS 33.310 [19], Annex E. The only difference is that TLS cipher suites without encryption shall not be used.

The certificates shall comply with the requirements for TLS certificates in clause 6.1 of TS 33.310 [19].

Support of certificate revocation and of the related fields in certificates is optional. If supported, the certificate and CRL profiles in clause 6.1 and 6.1a of TS 33.310 [19] should be followed.

NOTE 1: The management of Root Certificates is out of scope of this Technical Specification.

NOTE 2: If no revocation of certificates is deployed, it should be noted, however, that choosing short lifetimes for BSF certificates may considerably reduce the risk, in case BSF certificates may ever be compromised.

I.6.1 void

I.6.2 Authentication of the BSF

The Client shall authenticate the BSF by use of a server certificate. The client shall match the server name as specified in RFC 2818 [18] section 3.1.

The ME shall use a preconfigured list of trusted root certificates for 2G GBA BSF server certificate validation. BSF server certificate validation shall not require manual user interaction.

NOTE: The risk of the UE using the root certificates associated with a compromised Certification Authority (CA) can be greatly reduced when the preconfigured list of trusted root certificates is restricted to a low number of CAs trusted by the operator, as opposed to the list of all root certificates in a browser"s key store.

I.6.3 Authentication of the UE

The BSF shall not request a certificate in a Server Hello Message from the UE. The BSF shall authenticate the UE as specified in clause I.5.2 of this specification.

I.6.4 Set-up of Security parameters

The TLS Handshake Protocol negotiates a session, which is identified by a Session ID. The Client and the BSF shall allow for resuming a session. The lifetime of a Session ID is subject to local policies of the UE and the BSF. A recommended lifetime is five minutes.

NOTE: If the BSF adheres to the recommended lifetime the UE can be certain to be able to resume the TLS session in case of bootstrapping re-negotiation.

Annex J (informative): Usage of USS with local policy enforcement in BSF

This Annex describes how the local policy enforcement in the BSF is used between the NAF and the BSF to control the key delivery to the NAF.

J.1 General

A BSF may have a local policy for zero or more NAFs where the policy for a NAF may state that subscriber's GUSS shall include one or more USSs identified by a GSID. In other words, for a particular NAF the BSF may require that one more USSs shall be present in subscriber's GUSS.

In general, there are two network elements where access control based on some local policy is enforced, i.e. NAF and BSF. Thus two phases with access control based on USSs have to be covered:

1) Access control within NAF for Ua requests: Whether the subscriber is allowed to access the service is decided in the NAF and possibly with the help of USSs. Upon receiving the B-TID from the UE, the NAF fetches the NAF specific shared key (Ks_(ext/int)_NAF) from the BSF, and optionally fetches the USSs, which typically contain NAF specific persistent user identities, and authorization flags. Based on a local policy in the NAF, which may include evaluating the contents of the USS, the NAF decides whether the subscriber is allowed to access the service.

2) Access control within BSF for Zn requests: In certain cases, the operator may wish to implement access control in the BSF. This functionality can be used with any NAF, but the main reason for having this is to implement home operator control in the cases where the NAF is in a visited network.

This Annex describes the access control case within the BSF for Zn requests in more detail.

The following facts should be noted on use of this Annex:

- This access control is completely local to the network of the BSF operator (i.e. home operator of subscriber). This implies that no inter-operator agreement is necessary for implementation of this access control.

- The local policies of the BSF may be based on NAF names and on NAF groups. For the sake of brevity only NAFs are mentioned in the following descriptions.

J.2 Usage scenarios

Four different scenarios can be identified how the local policy enforcement in the BSF will work:

1) A NAF does not use USSs (i.e. it does not request a USS from the BSF), and the BSF does not have a local policy for this NAF.

2) A NAF does not use USSs (i.e. it does not request a USS from the BSF), and the BSF does have a local policy for this NAF.

3) A NAF does use USSs (i.e., it requests one or more USSs from the BSF), and the BSF does not have a local policy for this NAF.

4) A NAF does use USSs (i.e., it request one or more USSs from the BSF), and the BSF does have a local policy for this NAF.

The steps executed in each of these scenarios are described in more detail in the following subclauses.

In all scenarios the NAF has received B-TID from the UE over the Ua reference point before the following steps are executed. The steps describe only the procedures that are related to the local policy enforcement in the BSF with respect to USS existence. Also transfer of other information elements not related to this access control is not mentioned (e.g. key lifetime, private subscriber identity).

J.2.1 Scenario 1: NAF does not use USSs, BSF does not have local policy for NAF

In this scenario, the NAF does not use USSs and the BSF does not have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It does not include any GSIDs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is no local policy, i.e. for this particular NAF, the BSF does not require any USSs (identified by GSIDs) to be present in subscriber's GUSS.
4. The BSF derives the NAF specific shared key(s), and sends them to the NAF in the response.
5. The NAF receives the response with the NAF specific shared key(s).

After receiving the NAF specific shared key(s), the NAF may perform access control to the service according to its own policies and continues to communicate with the UE.

J.2.2 Scenario 2: NAF does not use USSs, BSF does have local policy for NAF

In this scenario, the NAF does not use USSs and the BSF does have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It does not include any GSIDs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is a local policy for this NAF, i.e. for this particular NAF, the BSF does not require any USSs (identified by GSIDs) to be present in subscriber's GUSS.

   The BSF checks whether all the required USSs identified by GSIDs are present in subscriber's GUSS: If yes, the BSF continues from step 4. If not, the BSF the BSF sends an error message to the NAF.

NOTE: As specified in clause 4.4.6, it is not required that the NAF requests the USSs over the Zn reference point, which the BSF requires to be present in the GUSS for particular NAF, rather it is sufficient that the BSF checks the presence of the USSs locally.

4. The BSF derives the NAF specific shared key(s), and sends them to the NAF in the response.
5. The NAF receives the response with the NAF specific shared key(s).

After receiving the NAF specific shared key(s), the NAF may perform access control to the service according to its own policies and continues to communicate with the UE.

If the NAF received the "not authorized" error message, it may indicate this to the UE over Ua reference point. In any case, the GAA based security setup will fail between the UE and the NAF since the NAF did not get the NAF specific shared key(s).

J.2.3 Scenario 3: NAF does use USSs, BSF does not have local policy for NAF

In this scenario, the NAF does use USSs and the BSF does not have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It includes the GSIDs it needs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is no local policy, i.e. BSF does not require USSs identified by GSIDs to be present in subscriber's GUSS.

4. The BSF derives the NAF specific shared key(s), and sends them and the USSs identified by the GSIDs to the NAF in the response. If a particular USS is not found in subscriber's GUSS, or the NAF is not authorized to receive a particular USS, these USSs are omitted from the response.

5. The NAF receives the response with the NAF specific shared key(s), and those requested USSs that were available (i.e., found in subscriber's GUSS and allowed by the BSF to be received by the NAF).

After receiving the NAF specific shared key(s) and the available USSs, the NAF may perform access control to the service according to its own policies (e.g. USS required or not, authorization flags required) and continue to communicate with the UE.

J.2.4 Scenario 4: NAF does use USSs, BSF does have local policy for NAF

In this scenario, the NAF does use USSs and the BSF does have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It includes the GSIDs it needs in the request.

2. The BSF locates the subscriber information in its local memory using the B-TID.

3. The BSF checks whether a local policy exists for the NAF - in this scenario there is a local policy for this NAF, i.e., one or more USSs identified by GSIDs shall be present in subscriber's GUSS.

   The BSF checks whether all the required USSs identified by GSIDs are present in subscriber's GUSS: If yes, the BSF continues from step 4. If not, the BSF the BSF sends an error message to the NAF.

4. The BSF derives the NAF specific shared key(s), and sends them and the USSs identified by the GSIDs to the NAF in the response. If a particular USS is not found in subscriber's GUSS, or the NAF is not authorized to receive a particular USS, these USSs are omitted from the response.

5. The NAF receives the response with the NAF specific shared key(s), and those requested USSs that were available (i.e., found in subscriber's GUSS and allowed by the BSF to be received by the NAF).

After receiving the NAF specific shared key(s) and the available USSs, the NAF may perform access control to the service according to its own policies (e.g. USS required or not, authorization flags required) and continue to communicate with the UE.

If the NAF received the "not authorized" error message, it may indicate this to the UE over Ua reference point. In any case, the GAA based security setup will fail between the UE and the NAF since the NAF did not get the NAF specific shared key(s).

Annex K (informative): Interoperator GBA-usage examples

This Annex gives examples how interoperator GBA is set up and operated.

K.1 Example on interoperator GBA setup

Interoperator GBA is set up the following way:

- Each home network operator sets up a BSF, which will enable bootstrapping sessions for its own subscribers.
- Each operator acting as a Serving Network for foreign subscribers in interoperator GBA needs to set up a Zn-Proxy which will forward the authentication requests from its own NAFs to the subscriber's home BSF outside of the VPLMN. The GBA secret is provisioned from the home operator's BSF through the Zn-Proxy to the NAF.

NOTE 1: The security requirements on the Zn' reference point between the Zn-Proxy and the BSF can be found in clause 4.2.2a.

- Each home operator that wants to provide the GBA secrets to foreign NAFs has to authorize these NAFs to request bootstrapping secrets. This is done by using TLS client certificates issued to Zn-Proxies in the serving network by the home network operator.

NOTE 2: The TLS client certificate profile is specified in the normative Annex E.

- An operator that wishes to co-operate in interoperator GBA with another operator shall issue a TLS client certificate to the other operator's Zn-Proxy. Two operators may both act as home operators or as serving operators (i.e., both possess a BSF and a Zn-Proxy), but this Annex also applies to configurations where one operator is always acting as home operator (i.e., hosts the BSF) and the other operator only as serving operator (i.e., the operator hosts only the Zn-Proxy). In the second case, where the serving foreign operator has the Zn-Proxy only, the TLS client certificate is to be handed down in one direction only (see also Annex E on usage of client certificates).

NOTE 3: The handling of TLS certificates is described in TS 33.310 [19]. When two operators sign a roaming agreement, they may also enrol TLS client certificate for each others Zn-Proxies.

NOTE 4: Interoperator GBA is based on bilateral agreements between the two operators. For example, if operator1 has a "GBA agreement" with operator2 and operator1 signs another "GBA agreement" with operator3, this does not mean that operator3 and operator2 have implicitly a "GBA agreement". Operator2 and operator3 shall separately sign a "GBA agreement".

NOTE 5: The home operator may use NAF groups to support local policy checks within its BSF (cf. clause 4.2.1). These may be e.g. one group for NAFs in home network and one group for NAFs in serving networks, or separate groups for each serving network the home operator has "GBA agreements" with. This NAF grouping is under sole responsibility of the home operator and only visible to him. The Zn-Proxies and NAFs in serving networks are not aware of any NAF grouping done in home network.

As described in clause 4.2.2a, a Zn-Proxy may be co-located with a BSF (see Figure K-2). This has the benefit that the NAF has only one logical channel to BSF/Zn-Proxy. Therefore the NAF does not need to make a decision based on the B-TID whether to send the authentication request to the Zn-Proxy or to the BSF. However, this decision can be based on the B-TID as it contains the address of the BSF.

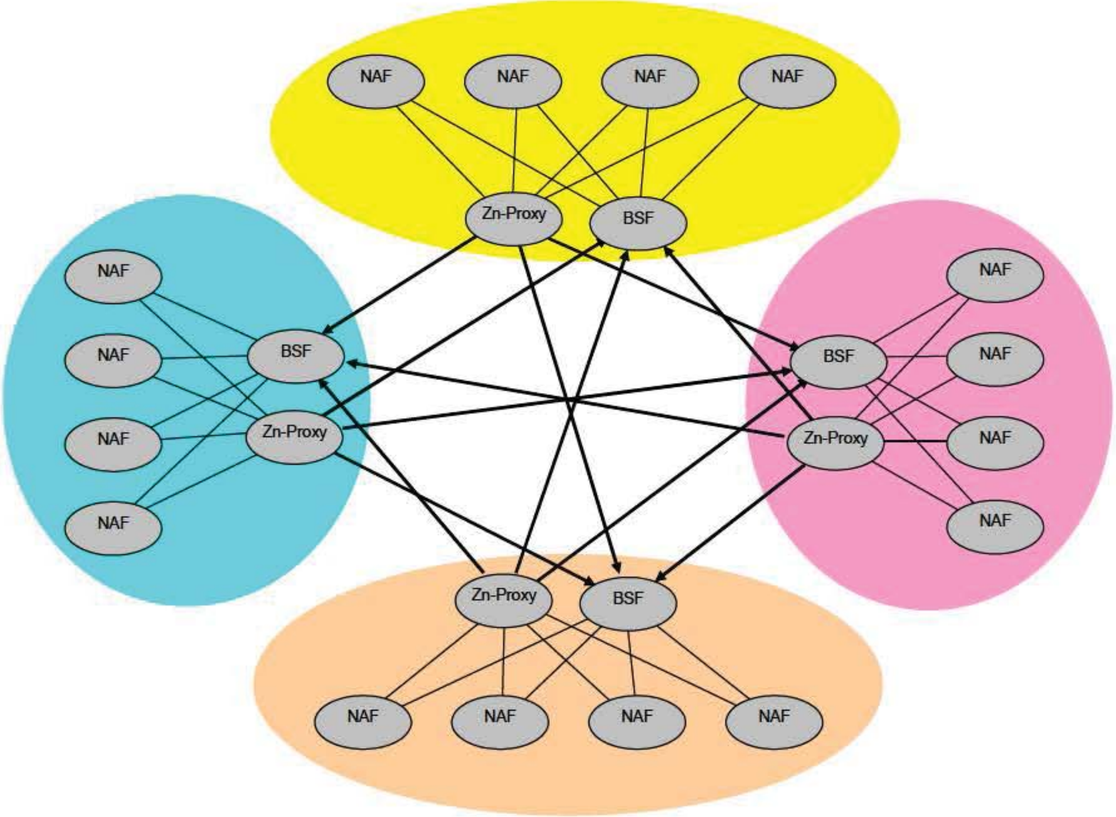


Figure K-1: Interoperator GBA with separate BSF and Zn-Proxy

NOTE 6: The figure K-1 does not show the most general case, where there could be one Zn-proxy per home network in each serving network. It is expected that networks will be optimized and that the existence of one dedicated Zn-proxy for each foreign subscriber home network will be a rare occurrence. The co-location of all Zn-Proxies of one serving network in one location as shown in Figure K-1 is a special case.

NOTE 7: The TLS connections between Zn-Proxy and BSF are "directed", this is indicated in Figure K-1 by the arrowed lines where the arrows point to the server TLS role. The role of the client certificates in these TLS connections is explicitly outlined. Each direction requires a TLS server certificate used at BSF and a TLS client certificate used at Zn-Proxy.

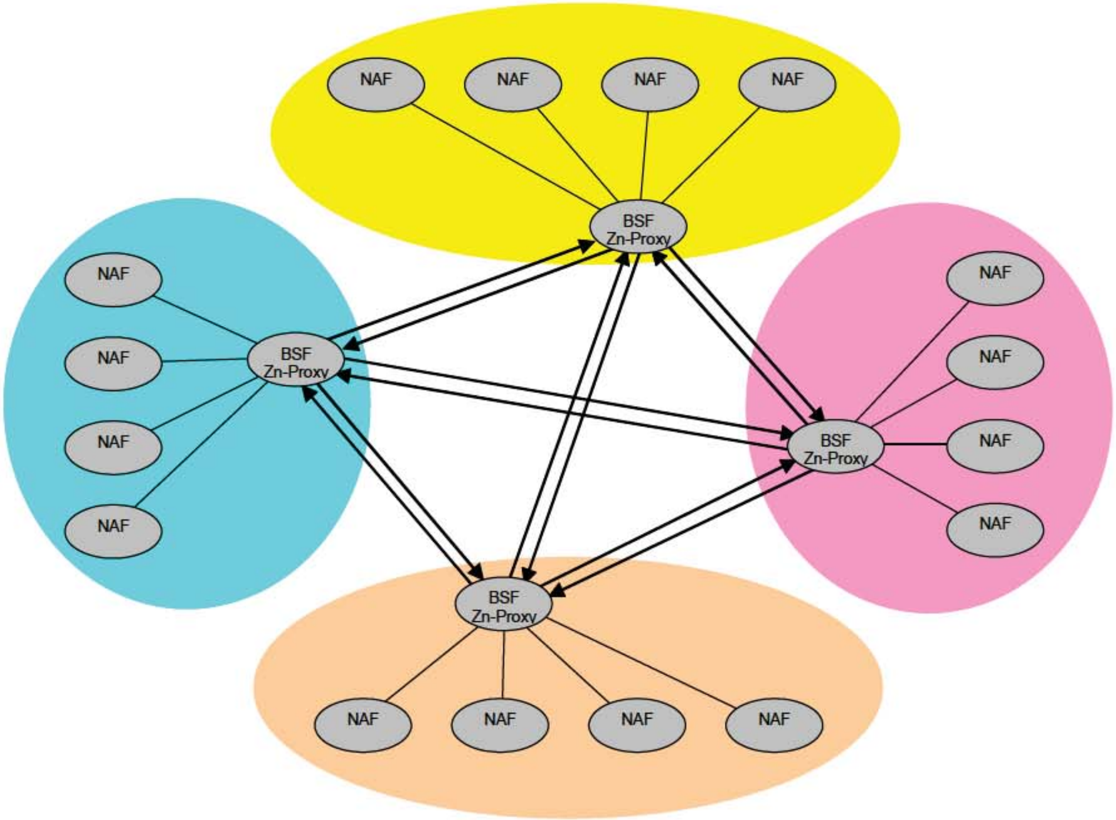


Figure K-2: Interoperator GBA with co-located BSF and Zn-Proxy

NOTE 8: The two distinct TLS connections between Zn-Proxy and BSF are "directed", this is indicated in Figure K-2 by the two lines. Thus the two TLS connection directions may not be intermixed, as the role of the client certificates in these TLS connections is explicitly outlined. Each direction requires a server TLS certificate used at BSF and a client TLS certificate used at Zn-Proxy.

K.2 Example on interoperator GBA operation

Interoperator GBA usage goes as follows:

NOTE 1: This description is based on GBA_ME bootstrapping to simplify the examples, but GBA_U bootstrapping can also be used in interoperator GBA operation.

1. A UE contacts a NAF that does not belong to subscriber's home network. The foreign NAF notifies the UE that 3GPP bootstrapping is required to secure the connection between the UE and the NAF.

2. The UE bootstraps with the home network via the subscriber's BSF. The address of subscriber's home BSF is generated from user's IMSI or IMPI as specified in TS 33.220, clause 4.5.4. The key Ks, and the B-TID are established between the BSF and the UE.

3. The UE derives the NAF specific key Ks_NAF, and uses Ks_NAF and the B-TID on the Ua reference point between the UE and the foreign NAF. At some point during this setup the UE transfers the B-TID to the NAF in the serving network.

4. Upon receiving the B-TID, the foreign NAF has two modes of operations depending on the actual setup of the Zn-Proxy and the BSF in the serving network:

NOTE 2: Any BSF in a network different from the home network of a subscriber and any Zn-Proxy are not visible to the subscriber. To avoid any confusion with the subscribers BSF in the home network, the BSF in a visited network is called foreign BSF in this clause.

a) If the Zn-Proxy and the foreign BSF are separate entities, the foreign NAF shall inspect the B-TID to discover whether the subscriber belongs to its own network, or whether it is a visiting subscriber. In the former case, the request for the Ks_NAF is sent to the BSF, in the latter case, the request is sent to the Zn-Proxy.

b) If the Zn-Proxy and the foreign BSF are a co-located entity, the NAF sends the request for the Ks_NAF to this co-located entity. The NAF does not need to inspect the B-TID.

NOTE 3: Since the B-TID contains the address of subscriber's home BSF, it can be used to discover the home network of the subscriber. A NAF supporting this approach can work with both separated and co-located configurations.

5. Upon receiving the request from the NAF, the Zn-Proxy shall inspect the following:

b) Validate that the NAF is authorized to request the Ks_NAF (i.e., the DNS part of NAF_Id in the message is correct).

b) Discover the BSF of the subscriber by inspecting the B-TID.

6. The Zn-Proxy will establish or use the existing DIAMETER or HTTP session to subscriber's home BSF. This DIAMETER or HTTP session is secured by TLS, and the Zn-Proxy shall use a client certificate that the BSF trusts.

7. The Zn-Proxy will forward the request to subscriber's home BSF.

8. Subscriber's home BSF shall validate that the DNS part of the NAF_Id in the request also exists in the client certificate of the Zn-Proxy.

9. Subscriber's home BSF locates the bootstrapping information using the B-TID, processes the request (including possible requests for USSs, local policy check, etc.), derive the NAF specific key, and send the response to the Zn-Proxy.

10. The Zn-Proxy will forward the response to the NAF.

11. The NAF continues with the Ua connection establishment with the UE.

Figure K-3 depicts the entities involved in the above procedure.

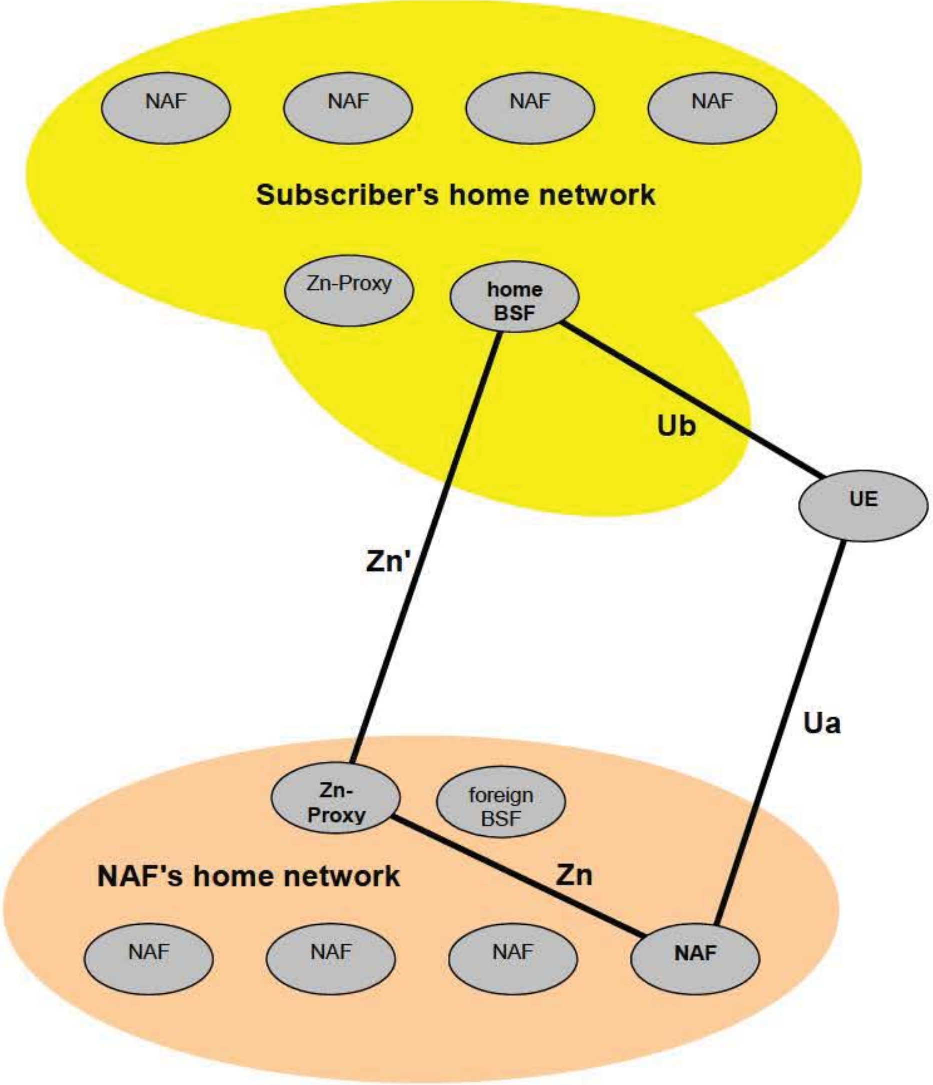


Figure K-3: Interoperator GBA usage

Annex L (informative): Information on how security threats related to known GSM vulnerabilities are addressed by the 2G GBA solution

The 2G GBA solution aims to provide mutual authentication between UE and BSF. This annex examines how the 2G GBA solution mitigates the impersonation of UE or the BSF i.e. security threats related to the known GSM vulnerabilities.

The threats that are originated from the weakness in the usage of the COMP128 algorithm exist independently of the usage of 2G GBA.

L.1 Impersonation of the UE to the BSF during the run of the Ub protocol

This is the main threat to the 2G GBA solution.

1) An attacker (being in the possession of 2G GBA equipment) could try to perform a Man-in-the-middle-attack, impersonating a genuine GSM user to the BSF. In this scenario the attacker would be at the client end of the TLS tunnel to the BSF and send the challenge RAND to the target GSM user, in order to obtain SRES and Kc. However, for the attack to be successful, he would have to find also Kc within the runtime allowed for steps 3 to 5 of the protocol over Ub, as specified in Annex I.5.2. This may be feasible when the terminal of the target GSM user still runs A5/2. A5/2 will be removed from networks by the end of 2006, and will not be present in any 2G GBA enabled terminals. A vulnerability caused by A5/2 would only exist in the case where a GSM user has subscribed to 2G GBA feature, but uses his SIM in an old terminal with A5/2 enabled while being targeted by the attacker. But the practical implications of this remaining vulnerability are expected to be limited as a user subscribed to 2G GBA will own a Release 7 terminal (2G GBA will be a Release 7 feature), and the likelihood of him inserting his SIM in an old terminal, and an attacker obtaining this information and exploiting it for a man-in-the-middle attack, may be low in practice. Furthermore, old terminals will gradually disappear.
   The attack may also be feasible when the attacker, using a false base station, forces the use of A5/1 on the ME. The attacker may then be able to determine Kc from the (encrypted) CIPHERING MODE COMPLETE message especially when the fillbits are not random. Note that the fillbits are required to be random from Rel-8 onwards, according to TS 44.006 [46].
   The attack may also be feasible when the attacker, using a false base station, forces the use of GEA1 on the ME and is able to determine Kc. Note that the implementation of GEA1 in MEs is forbidden from Rel-12 onwards, according to TS 43.020 [47].

2) SIM cloning: an attacker being able to find the long-term key Ki of a genuine GSM user is able to fully impersonate him in all contexts, including the 2G-GBA one (if this has been subscribed by the genuine user).. The attacker could do this by exploiting weaknesses of A3/A8 as they were found for COMP128, while in possession of the SIM i.e. the attacker tries to find the long term key K. Even if 2G GBA does not increase the risk of possible A3/A8 breakages, it has to be noted that the COMP128-related issue disappears when more secure A3/A8 algorithms are used. These are available today, cf. "GSM MILENAGE", as specified in TS 55.205 v610. Operators are advised in general to discontinue the use of COMP128

3) Unauthorized access to SIM needs to be countered by platform security methods. The impacts of a compromised SIM/ME or UICC/ME interface on GAA security are similar in 2G GBA and 3G GBA.

L.2 Impersonation of the BSF to the UE during the run of the Ub protocol

To prevent an impersonation attack of the BSF to the UE during the run of the Ub protocol the authentication of the BSF to the UE is improved by protecting the communication with TLS. An attacker succeeds only if he can break both, the certificate-based TLS authentication to the UE and mutual authentication provided by HTTP Digest using a password derived from GSM procedures. One way to break TLS is to compromise the certificate.

When an attacker was able to obtain a forged server certificate with the name of the genuine BSF from a compromised Certification Authority then the attacker could break the certificate-based TLS authentication to the UE. Furthermore, the attacker would be able to perform a make a man-in-the-middle attack between the UE and the BSF by playing TLS server towards the UE and TLS client towards the BSF. Such a a man-in-the-middle attack would make it possible for the attacker to read Ks-input and hence have a greater chance to compute the key Ks.

The man-in-the-middle attack could be countered by the use of channel binding as described in RFC 5929 [48]. This approach was not pursued further due to the perception that the risk posed by the relative weakness of GSM security was far greater than the risk posed by a CA.

NOTE: For a way of reducing the risk of the UE using the root key associated with a compromised Certification Authority (CA) see clause I.6.2 of the present specification.

L.3 Finding the GBA key Ks during or after the Ub protocol run

For BSF-to-UE authentication and for establishment of the key Ks, the solution relies on both, GSM security and TLS security. The attacker needs to know all the parameters of the GSM triplet, in particular Kc, and additionally break the TLS security, as the attacker also needs to know the Ks-input parameter confidentially transmitted by the BSF over TLS. Breaking GSM security after the Ub protocol run alone does not provide sufficient information to break 2G GBA.

L.4 Bidding down attack

To avoid a bidding down attack (also called downplay attack), the 2G GBA solution requires that a GBA-enabled terminal that supports SIM based 2G GBA must support also USIM/ISIM based 3G GBA as specified in I.2.4. If a USIM/ISIM is available, then the terminal must use the USIM/ISIM based 3G GBA as specified in I.4.8.

Annex M (normative): GBA_Digest

M.1 General

This annex specifies the use of SIP Digest credentials, as defined in TS 33.203 [16], for GBA. The procedure specified in this annex is called GBA_Digest. GBA_Digest allows access to applications in a more secure way than would be possible with the use of password-based HTTP Digest as specified in RFC 2617 [3] without enhancements. It may be useful for environments where a UICC, or a SIM card, is not available to subscribers. The use of GBA_ Digest is restricted to such environments.

Clauses 4 and 5 of the present document do not apply to this Annex unless explicitly stated.

NOTE: The use of the term 'UE' in this Annex is in line with the use of the term 'UE' in TS 33.203 [16], Annex N (on SIP Digest), but differs from that in other 3GPP specifications in that it assumes that a UICC is not available to subscribers in the UE.

M.2 Reference model

The reference model is the same as described in clause 4.1, with the exception that the reference point Zh' is not needed here.

M.3 Network elements

M.3.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using a combination of the HTTP Digest protocol and the TLS protocol, and agree on session keys that are afterwards applied between UE and a Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using the key derivation procedure as specified in Annex B. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause M.6.3.

The BSF shall be able to acquire the GBA user security settings (GUSS) from the HSS.

The BSF shall discover from the request received from the UE over the Ub interface whether the UE intends to run GBA_Digest. The BSF shall then request a SIP Digest authentication vector from the HSS or abort the Ub run with a suitable failure message, according to its local policy.

The BSF shall be able to keep a list, which assigns NAFs to NAF Groups. This list is used to select if any and which application-specific USS within GUSS is valid for a certain NAF.

NOTE 1: The operator does the assignment of NAFs to NAF Groups. NAF Group definitions in HSS and all connected BSFs belonging to the same operator's network shall be equal (cf., clause I.2.3). As these network elements belong to the same operator's network, standardisation of the NAF Group definitions themselves is not necessary in 3GPP.

NOTE 2: The NAF grouping may be e.g. "home" and "visited". It allows the BSF to send USSs for the same application with e.g. different authorization flags to different NAFs, e.g., in home network and visited networks. The NAF e.g. in visited network indicates only the requested application, but it is unaware of the grouping in home network of the subscriber.

The BSF shall allow the operator to configure a BSF policy whether to accept subscribers using SIP Digest credentials or not for a certain NAF.

M.3.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and a NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of a NAF are:

- there need not be a previous security association between the UE and the NAF;
- NAF shall locate and communicate securely with the subscriber's BSF;
- NAF shall acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire zero or more application-specific USSs from the HSS via the BSF;
- NAF shall be able to set the local validity condition of the shared key material according to the local policy;
- NAF shall be able to check lifetime and local validity condition of the shared key material;
- NAF shall have a policy whether to accept subscribers using SIP Digest credentials. However, whether GBA_Digest is allowed to be used with a specific Ua application or not, is dependent on the relevant Ua application. If there is a specific TS for an application using a particular Ua protocol, and unless this TS explicitly prohibits the use of GBA_Digest, the NAF may allow usage of SIP Digest credentials for this application,
- the NAF shall be able to indicate to the UE that the SIP Digest-based GBA bootstrapping security association is acceptable.

NOTE: Without additional measures, GBA, as defined throughout the present document, does not guarantee the freshness of the key, Ks_NAF, in the sense that it does not guarantee that the key was not used in a previous run of the Ua protocol. The additional measures which may be taken by the UE and the NAF to ensure key freshness in GBA are:

1) enforce a new run of the Ub protocol (thus generating a new Ks) before deriving a new Ks_NAF;

2) store previously used keys Ks_NAF, or the corresponding key identifiers B-TID, until the end of their lifetime.

A UE and a NAF that support a Ua protocol that does not provide replay protection over unconnected runs of the protocol, will need to take corresponding action to avoid replay attacks if desired.

M.3.3 Zn-Proxy

The text from clause 4.2.2a applies also here.

M.3.4 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS.

The requirements on the HSS are:

- HSS shall provide the only persistent storage for GUSSs;
- GUSS shall be defined in such a way that interworking of different operators for standardised application profiles is possible;
- GUSS shall be defined in such a way that profiles for operator specific applications and extensions to existing application profiles are supported without need for standardisation of these elements.
- GUSS shall be able to contain application-specific USSs that contain parameters that are related to identification or authorization information of one or more applications hosted by one ore more NAFs. Any other types of parameters are not allowed in the application-specific USS.

NOTE 1: The necessary subscriber profile data may be fetched by the NAF from its local database.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - subscriber specific key lifetime;
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or subscriber does not have GUSS then the BSF will use the default values in the BSF local policy defined by the particular MNO.

- HSS shall be able to assign application-specific USSs to a NAF Group. This shall be defined in such a way that different USSs for the same application, but for different groups of NAFs, are possible. The restrictions on the number of USSs per GUSS are dependent on the usage of NAF Groups by the operator:
  - if no NAF Groups are defined for this application then at most one USS per application is stored in GUSS;
  - if NAF Groups are defined for this application then at most one USS per application and NAF Group is stored in GUSS.
- NAF Group definitions in the HSS and all connected BSFs belonging to the same operator's network shall be equal.

- Information on UICC type and on key choice are not required for subscribers using SIP Digest credentials. GBA_Digest is regarded as ME-based.

M.3.5 UE

The required functionalities from the UE are:

- the support of HTTP Digest protocol according to RFC 2617 [3] with the additional profiling specified in this Annex;
- the support of TLS;
- the capability to use SIP Digest credentials in bootstrapping;
- the capability for a Ua application on the terminal to indicate to the GBA Function on the terminal whether SIP Digest credentials are allowed for use in bootstrapping;
- the capability to derive new key material to be used with the protocol over the Ua interface as defined in clause M.6.3;
- support of at least one Ua application protocol (For an example see TS 33.221 [5]);
- the capability to send an indication to the BSF over the Ub interface that the UE intends to run GBA_Digest.

M.3.6 SLF

The text from clause 4.2.5 applies also here.

M.4 Bootstrapping architecture and reference points

M.4.1 Reference point Ub

The reference point Ub is between the UE and the BSF. Reference point Ub provides mutual authentication between the UE and the BSF. It allows the UE to bootstrap the session keys based on SIP Digest credentials.

M.4.2 Reference point Ua

The reference point Ua carries the application protocol, which is secured using the keys material agreed between UE and BSF as a result of the run of the protocol over reference point Ub.

M.4.3 Reference point Zh

The reference point Zh used between the BSF and the HSS allows the BSF to fetch the required authentication information and all GBA user security settings from the HSS. The reference point Zh is an intra-operator domain interface.

M.4.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous protocol run over the reference point Ub from the UE to the BSF. It is also used to fetch application-specific user security settings from the BSF, if requested by the NAF.

M.4.5 Reference point Dz

The text from clause 4.3.5 applies also here.

M.5 Requirements and principles for bootstrapping

M.5.1 General Requirements

The following requirements and principles are applicable to bootstrapping procedure:

- the GBA_Digest bootstrapping function shall not depend on the particular NAF;
- the server implementing the bootstrapping function needs to be trusted by the home operator to handle authentication vectors;
- the server implementing the NAF needs only to be trusted by the home operator to handle derived key material;
- it shall be possible to support NAF in the operator's home network and in the visited network;
- the architecture shall not preclude the support of network application function in a third network;
- to the extent possible, existing protocols and infrastructure should be reused;
- in order to ensure wide applicability, all involved protocols are preferred to run over IP;
- it shall be prevented that a security breach in one NAF who is using the GBA, can be used by an attacker to mount successful attacks to the other NAFs using the GBA;
- an attacker shall not be able to exploit a security breach in one security protocol over Ua in order to mount a successful attack against a different security protocol over Ua;
- If USIM, ISIM, or SIM are available and the BSF supports AKA-based GBA the UE shall not use GBA_Digest. Instead, the UE shall use the procedures as specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall not impact the procedures for AKA-based GBA as specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall not reduce security for users of AKA-based GBA;
- GBA_Digest shall be closely modelled after AKA-based GBA specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall provide measures to mitigate known vulnerabilities of the re-use of SIP Digest credentials.

M.5.2 Access independence

The bootstrapping procedure for GBA_Digest is, in principle, access independent as it only requires IP connectivity from the UE. However, in order to ensure that GBA_ Digest is not used over access networks defined in 3GPP specifications operators may introduce some access dependence in their network configurations, e.g. by assigning different ports on the BSF to different access networks.

M.5.3 Authentication methods

Authentication between the UE and the BSF shall not be possible without a valid IMS subscription. Authentication shall be based on a combination of the HTTP Digest protocol using SIP Digest credentials and the TLS protocol, as defined in clause M.6.3. TLS shall be used with server certificates, but the TLS server shall not request client certificates.

M.5.4 Roaming

The requirements on roaming are:

- A subscriber located outside the home network shall be able to utilize the bootstrapping function in the home network. The subscriber shall be able to utilize a network application function that is outside the home network.

- The home network shall be able to control whether its subscriber is authorized to use the service outside the home network.

M.5.5 Requirements on reference point Ub

The requirements for reference point Ub are:

- the BSF shall be able to identify the UE;

- the BSF and the UE shall authenticate each other based on the methods specified in clasue M.5.3;

- the BSF shall send a bootstrapping transaction identifier to the UE;

- the UE and the BSF shall establish shared keys;

- the BSF shall indicate to the UE the lifetime of the key material. The key lifetime sent by the BSF over Ub shall indicate the expiry time of the key.

NOTE: This does not preclude a UE to refresh the key before the expiry time according to the UE's local policy.

M.5.6 Requirements on reference point Zh

The requirements for reference point Zh are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: This requirement may be fulfilled by physical or proprietary security measures since BSF and HSS are located within the same operator's network.

- the BSF shall send a bootstrapping information request concerning a subscriber;

- optionally the BSF may have the capability to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);

- the HSS shall send one SIP Digest authentication vector at a time to the BSF;

- the HSS shall send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

NOTE 2: If subscriber's GUSS is updated in HSS, this is not propagated to the BSF. The GUSS in the BSF is updated when the BSF next time fetches the authentication vectors and GUSS from the HSS over Zh reference point as part of the bootstrapping procedure.

- no state information concerning bootstrapping shall be required in the HSS;

- all procedures over reference point Zh shall be initiated by the BSF;

- the number of different interfaces to the HSS should be minimized.

M.5.7 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;

- If the BSF and the NAF are located within the same operator's network, the DIAMETER based Zn reference point shall be secured according to NDS/IP in TS 33.210 [13];

- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E;
- The BSF shall verify that the requesting NAF is authorised to obtain the key material or the key material and the requested USS;
- The NAF shall send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The NAF shall indicate to the BSF for each Zn run whether it is willing to accept Ks_NAF based on GBA_Digest;
- The BSF shall send the requested key material to the NAF;
- The NAF shall be able to get a selected set of application-specific USSs from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall indicate to the BSF the single application or several applications it requires USSs for;

NOTE 1: If some application needs only a subset of an application-specific USS the NAF selects this subset from the complete set of USS sent from BSF.

- The BSF shall be configured on a per NAF or per application basis if private subscriber identity and which application-specific USSs may be sent to a NAF;

NOTE 2: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI (but then there is no user anonymity). If the BSF does not send the IMPI or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- If a NAF requests USSs from the BSF and they are not present in subscriber's GUSS, it shall not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF shall then send only the requested and found USSs to the NAF;
- It shall be possible to configure a local policy as follows: BSF may require one or more application-specific USS to be present in a particular subscriber's GUSS for a particular requesting NAF, and to reject the request from the NAF in case the conditions are not fulfilled. In order to satisfy this local policy, it is not required that the NAF requests the USSs over the Zn reference point, which the BSF requires to be present in the GUSS, rather it is sufficient that the BSF checks the presence of the USSs locally. It shall also be possible configure the BSF in such a way that no USS is required for the requesting NAF;
- The BSF shall indicate to the NAF the bootstrapping time and the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 3: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 4: If one or more of the USSs that have been delivered to the NAF has been updated in subscriber's GUSS in the HSS, this change is propagated to the NAF the next time it fetches the USS from the BSF over Zn reference point (provided that the BSF has updated subscriber's GUSS from the HSS over Zh reference point).

- If the NAF indicated its willingness to accept Ks_NAF based on GBA_Digest in the Zn request and the B-TID sent by the NAF points to a Ks generated by GBA_Digest the BSF shall send information to the NAF that the subscriber is a subscriber who used SIP Digest credentials. If the B-TID points to a Ks established by another GBA method the BSF shall respond according to that method. Otherwise, the BSF shall not send key material to the NAF.

NOTE 5: This requirement enables a NAF to accept subscribers using SIP Digest credentials according to its local policy. The second sentence ensures backward compatibility with the procedures specified in clauses 4 and 5 and Annex I. Note also that inclusion of information on the GBA variant in the GUSS is not possible as one subscriber may have both AKA and SIP Digest credentials, leading to a depencence on the credentials actually used during the last Ub run.
A NAF that can understand a 'GBA_Digest' indication received from the BSF on Zn can understand which GBA variant was used on Ub to derive the Ks_NAF key and, hence, can always make its own judgment whether to accept the Ks_NAF based on its local policy. However, there is no technical reason why the NAF would not accept a Ks_NAF that was derived using an AKA-based GBA variant because such a Ks_NAF is stronger than a key that was derived using GBA_digest and there is no difference in using it for the NAF.

- The BSF may determine according to its local policy that the NAF shall not serve subscribers using SIP Digest credentials. If this is the case, the BSF shall not send keys generated by GBA_Digest to the NAF.

NOTE 6: This requirement allows an operator controlling the BSF to determine which applications shall use AKA-based GBA only.

- The NAF shall indicate to the BSF the protocol identifier of Ua security protocol for which it requires the key material by sending NAF-Id to BSF (cf. Annex H).

M.5.8 Requirements on Bootstrapping Transaction Identifier

The text from clause 4.4.7 applies also here.

M.5.9 Requirements on reference point Ua

The text from clause 4.4.9 applies also here.

M.5.10 Requirements on reference point Dz

The text from clause 4.4.10 applies also here.

M.5.11 Requirements on GBA keys and parameters handling

- The terminal shall delete all GBA keys related to a certain Ks (i.e., Ks itself, and NAF specific keys derived from this specific Ks) and the corresponding NAF_IDs, B-TID, , Ks lifetime, and, if applicable, Ks_NAF lifetimes and lifetimes of the keys derived from a Ks_NAF, when the key lifetime of this specific Ks expires.

M.6 Procedures

M.6.1 General

This chapter specifies in detail the format of the GBA_Digest bootstrapping procedure that is further utilized by various applications. It contains the authentication procedure with BSF, and the key material generation procedure.

M.6.2 Initiation of bootstrapping

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use GBA. When a UE wants to interact with a NAF, but it does not know if the NAF requires the use of shared keys obtained by means of the GBA, the UE may contact the NAF for further instructions (see figure M.1).

NOTE: The above text implies that a UE may contact either the BSF or the NAF without knowing whether the NAF supports GBA.

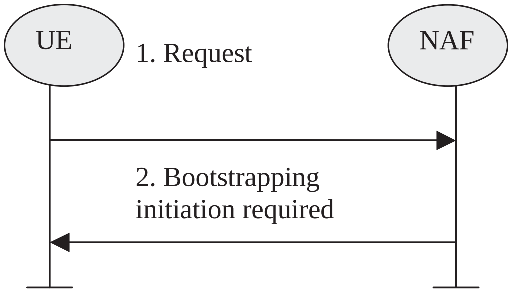


Figure M.1: Initiation of bootstrapping

1. The UE may start communication over reference point Ua with the NAF with or without any GBA-related parameters.

2. If the NAF requires the use of shared keys obtained by means of the GBA, but the request from UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message. If the use of GBA_Digest is acceptable to the NAF the NAF shall indicate it in this message. The form of this initiation message may depend on the particular reference point Ua.

M.6.3 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform such a procedure. Otherwise, the UE shall perform a bootstrapping procedure only when it has received a bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. clause M.6.4).

The bootstrapping procedure using SIP Digest credentials is run over the Ub interface (extended for the purposes of GBA_Digest) as described below:

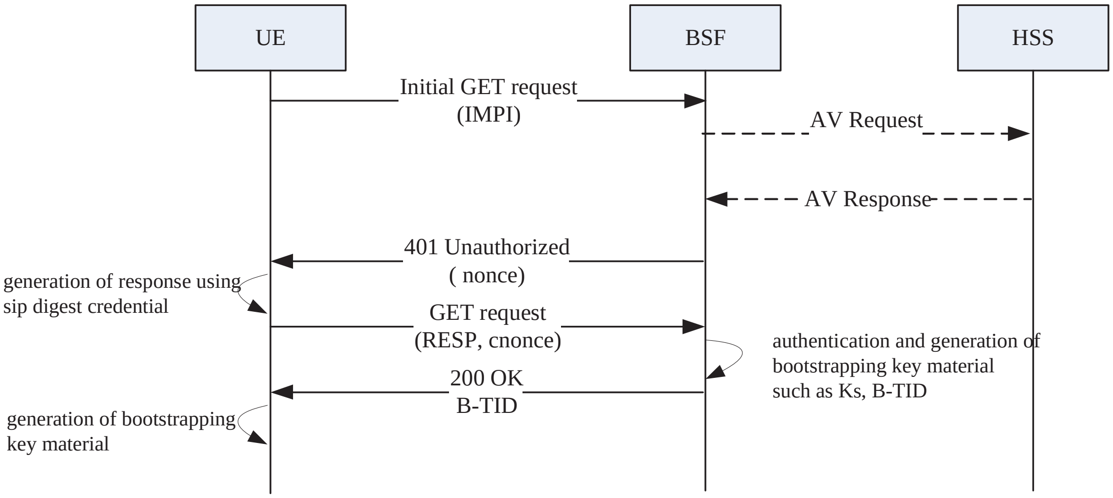


Figure M.2 GBA_Digest bootstrapping procedure

NOTE 1: Figure M.2 only shows an example flow for visualization and not all details are included.

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when sending HTTP messages over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when sending HTTP messages over Ub.

NOTE 1a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

Step 0:

The UE and the BSF shall establish a TLS tunnel with server authentication using a server certificate. The use of TLS message integrity is mandatory, while the use of TLS encryption is optional. All further messages between the BSF and UE shall be sent through this tunnel.

NOTE 2: TLS encryption can be useful for protecting the user identity privacy when the TMPI mechanism defined in the present document is not used.

Step 1:

In this HTTP request message from the UE to the BSF, the UE shall include an Authorization header containing a user identity in the "username" parameter and a token indicating the use of GBA_Digest. When a TMPI associated with the IMPI in use is available on the UE, this user identity shall be this TMPI, otherwise it shall be the IMPI. The realm in the Authorization header shall be the realm as defined for SIP Digest in TS 33.203 [16].

Step 2:

The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF shall look up the corresponding IMPI in its local database. If the BSF does not find an IMPI corresponding to the received TMPI it shall return an appropriate error message to the UE. The UE shall then delete the TMPI and retry the request using the IMPI.The BSF shall request a SIP Digest Authentication

Vector (SD-AV) from the HSS. The SD-AV is defined in TS 33.203 [16], Annex N. The username field in the Multimedia Auth Request shall contain the IMPI.

Step 3:

The HSS shall retrieve the SD-AV corresponding to the IMPI and send it to the BSF in a Multimedia Auth Answer. The handling of GUSS between BSF and HSS shall be as described in clause 4.5.2, step 2.

The qop value shall be set to "auth-int ".

NOTE 3: The additional protection afforded by qop set to "auth-int" may seem unnecessary considering the fact that the messages exchanged between UE and BSF are protected by a TLS tunnel. However, the use of "auth-int" is consistent with the other modes of GBA (GBA_ME, GBA_U and 2G GBA) and also provides a second layer of integrity protection in case the TLS server authentication is ever compromised (e.g. due to replacement of insecurely stored root certificates on the UE or a Certification Authority being compromised).

Step 4:

In the HTTP 401 Unauthorized response from the BSF to the UE, the BSF shall include a WWW-Authenticate header with parameters as specified in RFC 2617 [3].

The parameters realm, qop, and algorithm were provided in the SD-AV in step 3 and the nonce=base64encode (16 byte random value) is generated according to RFC 3548 [12] by the BSF.

Step 5:

When responding to a challenge from the BSF, the UE shall generate a cnonce randomly, and calculate the response RESP. The RESP shall be put into the Authorization header and sent back to the BSF in the GET request.

RESP shall be computed as a Digest-response according to RFC 2617 [3] (HTTP Digest) from the most recent GBA_Digest challenge and a password 'passwd' that is generated as follows:

passwd = KDF (H(A1), "GBA_Digest_RESP", TLS_MK_Extr)

where H(A1) is the hash of the following three parameters: the user name and password used by the user in IMS for SIP Digest according to TS 33.203 [16], Annex N, and the realm, cf. also RFC 2617 [3]. "GBA_Digest_RESP" is a character string. TLS_MK_Extr is extracted from the TLS master key according to RFC5705 [44] with the optional context value being omitted, the label set to "EXPORTER_GBA_Digest", and the length set equal to the length of the TLS master secret (48 bytes). KDF is the key derivation function as specified in clause B.2.

NOTE 4: A cautionary note on notation: According to RFC 2617 [3], the computation of RESP from the password 'passwd' defined above entails again a parameter called H(A1). This parameter will differ from the value of H(A1) that is input to the above formula because the passwords from which these two H(A1) values are derived differ. But no new notation is deemed necessary here as the notation H(A1), when H(A1) is derived from 'passwd', is not explicitely used in the text of the present document.

Step 6:

Upon receiving a GET request carrying the authentication response RESP, the BSF shall check that the expected RESP (calculated by the BSF in the same way as by the UE in step 5) matches the received RESP. If the check is successful then the user has been authenticated.

The BSF shall then derive Ks as follows, (see clause B.5 for the formation of the input):

Ks = KDF (H(A1), "GBA_Digest_Ks", TLS_MK_Extr, RESP)

where H(A1), RESP, and TLS_MK_Extr are defined as in step 5, and "GBA_Digest_Ks" is a character string.

The BSF shall generate the bootstrapping transaction identifier (B-TID) for the IMPI and store the tuple <B-TID, IMPI, Ks, nonce>. The B-TID shall be constructed in the format of a NAI by taking the nonce from step 4, and the BSF server name, i.e. nonce@BSF_server_domain_name.

NOTE 5: The B-TID construction above is almost identical to the one used in clause 4. The difference is that in clause 4 the username part is constructed from the (base64 encoded) RAND value.

The BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

NOTE 6: The formulations in the preceding paragraph, and the corresponding paragraph below relating to the computation of the TMPI in the UE, differ from the ones in clause 4.5.2 as GBA_Digest-aware UEs and BSFs always include the product tokens as described at the start of this clause. So, the condition in clause 4.5.2 is not needed.

The BSF shall send a 200 OK response to the UE to indicate the success of the authentication.

In this message from the BSF to the UE, the BSF shall include the bootstrapping transaction identifier (B-TID) and the key lifetime.

An Authentication-Info header according to RFC 2617 [3] shall be included into the 200 OK response.

The UE shall abort the procedure if the server authentication according to RFC 2617 [3] fails. Otherwise, the UE shall derive Ks in the same way as the BSF did above.

The UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

After successful bootstrapping procedure the UE and the BSF shall store the key Ks, the nonce, the B-TID, and an indication of the underlying security quality, i.e. GBA_Digest, for further use, until the key Ks is updated or until the deletion conditions in clause M.5.11 are satisfied. The key Ks shall then be used in the BSF and in the UE to derive NAF specific key(s) Ks_NAF to secure Ua reference points in the following way:

> Ks_NAF shall be computed as Ks_NAF = KDF (Ks, "gba-digest", nonce, IMPI, NAF_Id), where KDF is the key derivation function as specified in clause B.2, and the input parameters consist of the user's IMPI, the NAF_Id and 'nonce'. 'nonce' is the nonce that was used for computing the RESP that was input to the derivation of Ks. The NAF_Id shall be constructed as in clause 4.5.2. The "gba-digest" parameter is a static character string.

NOTE 6: The above derivation of Ks_NAF is analagous to the derivation in clause 4.5.2, step 9, and the same KDF can be utilized.

The KDF shall be implemented in the terminal.

M.6.4 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause M.6.2.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with an NAF the following steps are executed as depicted in figure M.3.

1. UE starts communication over reference point Ua with the NAF:

   - in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, and if the use of a Ks derived from an AKA-based GBA variant according to clauses 4.5.3, 5.5.3, or I.5.3, is not possible, the UE proceeds as follows:

     - if the UE knows (through a lack of indication in the Initiation of Bootstrapping procedure or by configuration) that the use of GBA_Digest is not acceptable to the NAF it shall abort the communication with the NAF. Otherwise, a key Ks_NAF shall be derived in the following way:

     - if a key Ks derived from SIP Digest credentials is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause M.6.3;

  - if no key Ks derived from SIP Digest credentials is available in the UE, the UE first agrees on a new key Ks derived from SIP Digest credentials with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

NOTE 0: A key Ks derived from an AKA-based GBA variant could still be available from a previous GBA bootstrapping run where the UICC was available, and could then still be used.

If it is not desired by the UE to use the same Ks derived from SIP Digest credentials to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF;

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g. because the key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure M.4. If the key's lifetime has expired the protocol used over reference point Ua shall be terminated. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause M.6.3, in order to obtain a new key Ks.

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause M.6.3). For each protocol used over Ua it shall be specified if only cases (1) and (2) of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 1: If the shared key between UE and NAF is invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

- the UE supplies the B-TID to the NAF, in the form as specified in clause M.5.8, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 2: The UE may adapt the key material Ks_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of the present document.

- the key management procedures for GBA related keys in the terminal are described in section M.5.11.

- when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

  According to the procedures defined in clauses M.6.3 and M.6.4, in the UE there is at most one Ks_NAF key stored per NAF_Id.

2. NAF starts communication over reference point Zn with BSF:

   - The NAF shall request key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua. The NAF shall indicate to the BSF whether it is willing to accept Ks_NAF based on GBA_Digest;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF_Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

3. The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause M.6.3, and supplies to NAF the requested key Ks_NAF, as well as the bootstrapping time and the lifetime of that key, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. In addition, the BSF shall indicate to the NAF that the subscriber is a subscriber using SIP Digest credentials. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

NOTE 4: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

NOTE 5: The NAF will adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of the present document.

- If the NAF did not indicate that it is willing to accept a Ks_NAF based on GBA_Digest, or if the BSF determines according to its local policy that the NAF shall not serve subscribers using SIP Digest credentials, then the BSF shall not send a Ks_NAF based on GBA_Digest;

- If the NAF indicated that it is willing to accept a Ks_NAF based on GBA_Digest, but the B-TID refers to a key Ks established by using an AKA-based method, then the BSF shall send a key Ks_NAF derived from this Ks unless this Ks was derived from 2G GBA and the NAF does not accept 2G GBA (cf. NOTE 0);

- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause M.5.7). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF;

- The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy;

- If the NAF determines, according to its local policy, that the NAF shall not serve subscribers using SIP Digest credentials, the NAF shall terminate the protocol over the reference point Ua;

- The NAF should accept the Zn response even when the GBA_Digest indication is missing (as this means that the key Ks_NAF was derived from a key Ks established by using an AKA-based method, which is stronger), (cf. NOTE 0);

- When the NAF receives the Zn response, it shall check that the GBA type in the Zn response corresponds with the GBA type negotiated over Ua protocol. If this is not the case, NAF shall terminate the protocol over the reference point Ua.

4. NAF continues with the protocol used over the reference point Ua with the UE.

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way.

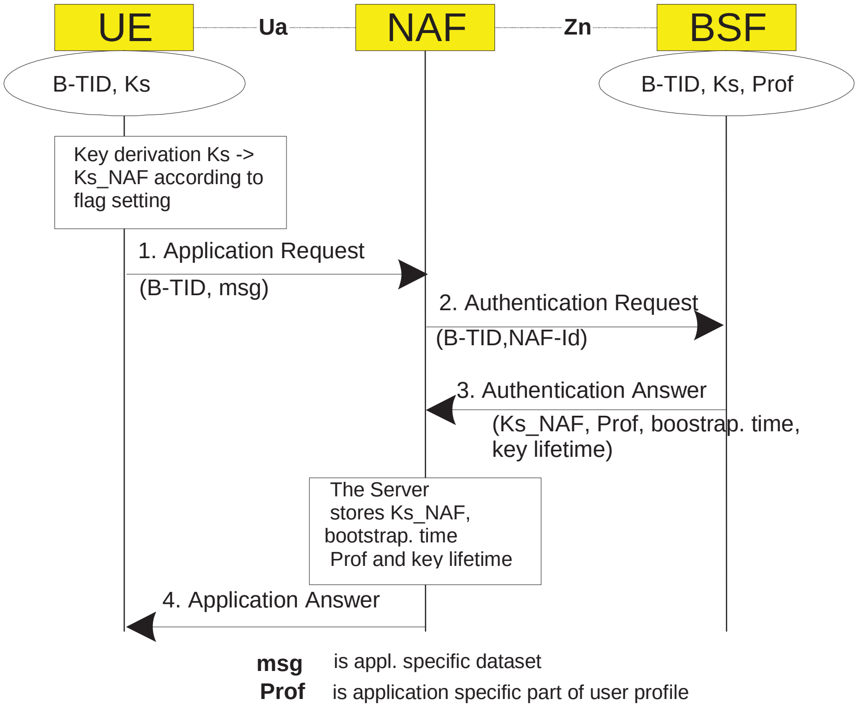

UE
Ua
NAF
Zn
BSF
B-TID, Ks
B-TID, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(B-TID, msg)
2. Authentication Request
(B-TID,NAF-Id)
3. Authentication Answer
(Ks_NAF, Prof, boostrap. time, key lifetime)
The Server stores Ks_NAF, bootstrap. time Prof and key lifetime
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure M.3: The bootstrapping usage procedure

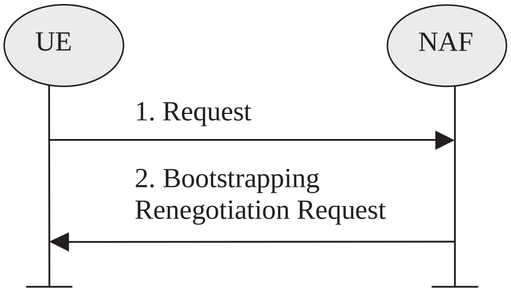


Figure M.4: Bootstrapping renegotiation request

M.6.5 Procedure related to service discovery

The UE shall discover the address of the BSF from the IMPI related to the IMS subscription. When the IMPI was derived from an IMSI as defined in clause 13 of TS 23.003 [11] then the BSF address shall be derived as as specified in clause 16 of TS 23.003 [11] for the case of an IMSI, otherwise the BSF address shall be derived as as specified in clause 16 of TS 23.003 [11] for the case of an IMPI.

NOTE: The reason for this distinction is the NOTE in clause 16 of TS 23.003 [11] warning that BSF addresses of a certain form may be unreachable.

M.7 TLS Profile

M.7.1 General

The UE and the BSF shall support TLS according to the TLS profile given in TS 33.310 [19], Annex E.

The certificates shall comply with the requirements for TLS certificates in clause 6.1 of TS 33.310 [19].

Support of certificate revocation and of the related fields in certificates is optional. If supported, the certificate and CRL profiles in clauses 6.1 and 6.1a of TS 33.310 [19] should be followed.

NOTE 1: The management of Root Certificates is out of scope of the present document.

NOTE 2: If no revocation of certificates is deployed, it should be noted, however, that choosing short lifetimes for BSF certificates may considerably reduce the risk, in case BSF certificates may ever be compromised.

M.7.2 Authentication of the BSF

The Client shall authenticate the BSF by use of a server certificate. If the BSF also supports 2G GBA under the same BSF address (cf. TS 23.003 [11]) it is recommended that this certificate be the same as the one used for BSF authentication in 2G GBA, cf. clause I.6.2. The client shall match the server name as specified in RFC 2818 [18], section 3.1.

NOTE: If the BSF addresses derived for the 2G GBA case and the GBA_Digest case differ, but the BSF is the same, then the operator can issue a BSF certificate with two server names or with a wildcard server name to ensure the client check of the server name works correctly, or the operator can issue two different BSF certificates.

The terminal shall use a preconfigured list of trusted root certificates for GBA_Digest BSF server certificate validation. It is recommended that this list be the same as the one used for 2G GBA BSF server certificate validation, cf. clause I.6.2. BSF server certificate validation shall not require manual user interaction.

M.7.3 Authentication of the UE

The BSF shall not request a certificate in a Server Hello Message from the UE. The BSF shall authenticate the UE as specified in clause M.6.3.

M.7.4 Set-up of Security parameters

The TLS Handshake Protocol negotiates a session, which is identified by a Session ID. The Client and the BSF shall allow for resuming a session. The lifetime of a Session ID is subject to local policies of the UE and the BSF.

Annex N (informative): Change history

Change history									
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**	**WI**
2004-03	SP-23	SP-040175	-	-	D	Presented for approval at TSG SA #23	1.2.1	2.0.0	
2004-03	SP-23	-	-	-	F	Approved and placed under Change Control (Rel-6)	2.0.0	6.0.0	
2004-06	SP-24	SP-040375	001	-	F	Removal of Annex A	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040376	002	-	B	NAF remove the security associations	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040377	003	1	D	Removal of editors notes on Transaction Identifiers	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040378	004	1	B	Introduction of a UICC-based Generic Bootstrapping Architecture	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040379	005	-	D	Editorial corrections to TS 33.220	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040380	006	-	C	Support for NAF in visited network	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040381	007	-	C	Editorial changes and clarifications to TS 33.220	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040382	008	-	F	Multiple key derivation mandatory	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040383	009	-	C	NAF's public hostname verification	6.0.0	6.1.0	SEC1--SC
2004-09	SP-25	SP-040619	010	-	C	Detailing of key lifetime	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	011	-	C	Details of USIM/ISIM usage in GAA	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	012	-	C	Generic Ua interface requirements	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	013	-	F	B-TID generation	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	014	-	B	Securing Zn reference point	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	015	-	D	GBA User Security Settings	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	016	-	C	Creation of GBA U AV in the BSF	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	017	-	F	Clarification of the definition of a default type of NAF-specific key	6.1.0	6.2.0	SEC1-SC
2004-12	SP-26	SP-040855	018	1	C	BSF discovery using default domain method	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	019	1	C	Local validity condition set by NAF	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	020	3	C	GBA User Security Settings (GUSS) usage in GAA and Introduction of NAF groups	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	021	2	C	Details of USIM/ISIM selection in GAA	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	023	-	B	TLS profile for securing Zn' reference point	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	025	2	F	Optimization of the GBA U key derivation procedure	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	027	2	F	Requirement on ME capabilities for GBA U	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	034	1	D	Adding a note about replay protection	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	035	1	C	Complete the MAC modification for GBA U	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	036	1	F	Removal of unnecessary editor's notes	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	038	1	C	Fetching of one AV only on each Zh run between BSF and HSS	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	039	1	B	Clean up of TS 33.220	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	040	1	F	New key management for ME based GBA keys	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	041	1	C	Key derivation function	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	042	1	D	Re-negotation of keys	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	043	1	C	No GUSS/USS update procedures in Release-6	6.2.0	6.3.0	GBA-SSC
2004-12	SP-26	SP-040855	044	1	C	Clarify the number of NAF-specific keys stored in the UE per NAF-Id	6.2.0	6.3.0	SEC1-SC
2005-03	SP-27	SP-040139	045	1	F	Key derivation function: character encoding	6.3.0	6.4.0	SEC1-SC
2005-03	SP-27	SP-040139	047	1	D	Bootstrapping timestamp	6.3.0	6.4.0	SEC1-SC
2005-03	SP-27	SP-040139	048	-	F	Storage of B-TID in GBA U NAF Derivation procedure	6.3.0	6.4.0	SEC1-SC
2005-06	SP-28	SP-050262	050	1	F	Usage of USS for local policy enforcement in BSF	6.4.0	6.5.0	SEC1-SC
2005-06	SP-28	SP-050262	051	1	F	Correcting figure 4.4	6.4.0	6.5.0	SEC1-SC
2005-06	SP-28	SP-050263	052	-	B	GBA User Security Settings (GUSS) transfer optimisation	6.4.0	7.0.0	SEC1-SC
2005-09	SP-29	SP-050553	0054	-	A	Clarification of anonymous access to NAF in GBA	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050554	0056	-	A	Removing IMPI from USS	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050572	0057	-	C	Informative annex on usage of USS for local policy enforcement in BSF	7.0.0	7.2.0	SEC1-SC
2005-09	SP-29	SP-050557	0059	-	A	Removing duplication of text relating to BSF addressing	7.0.0	7.1.0	SEC-SC1
2005-09	SP-29	SP-050555	0061	-	A	Clarification of lifetime of derived keys	7.0.0	7.1.0	SEC1-SC

2005-09	SP-29	SP-050575	0062	-	B	Introduction of key selection mechanism	7.0.0	7.2.0	SEC1-SC
2005-09	SP-29	SP-050556	0064	-	A	Addition of the Dz interface for multiple HSS deployments	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050565	0066	-	A	Removing requirement to send authentication vectors in batches	7.0.0	7.1.0	GBA
2005-09	SP-29	SP-050551	0068	-	A	Clarification concerning input parameter encoding for GBA_U NAF derivation procedure	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050577	0069	1	B	Normative annex on 2G GBA	7.0.0	7.1.0	2G GBA
2005-09	SP-29	SP-050552	0071	-	A	Providing Ua-security protocol based key separation	7.0.0	7.1.0	SEC1-SC
2005-10	post SP-29	-	-	-	-	Editorial change to align annexes between Release 6 and Release 7 based on CRs at SP-29	7.1.0	7.1.1	
2005-12	SP-30	SP-050768	0073	-	A	NAF_Id encoding	7.1.1	7.2.0	SEC1-SC
2005-12	SP-30	SP-050775	0074	-	B	Informative annex with examples on interoperator GBA usage	7.1.1	7.2.0	GAA2
2005-12	SP-30	SP-050775	0075	-	F	Clarification of local policy enforcement	7.1.1	7.2.0	GAA2
2005-12	SP-30	SP-050777	0076	-	F	Alignment of 2G GBA with recent CRs	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050777	0077	-	F	Addition of information requested by SA plenary on 2G GBA	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050777	0078	-	F	IMPI obtained from IMSI in 2G GBA	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050775	0079	-	F	Removal of possible interoperability problems	7.1.1	7.2.0	GAA2
2006-03	SP-31	SP-060061	0080	-	B	D-proxy renaming to Zn-Proxy	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060061	0082	-	B	Protection of Zn/Zn' reference point for http based protocols	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060051	0084	-	A	Restricting the TLS CipherSuites in Annex E and cleanup of references	7.2.0	7.3.0	SEC1-SC
2006-03	SP-31	SP-060061	0085	-	F	Clarifications of requirement	7.2.0	7.3.0	SEC7-2GGBA
2006-03	SP-31	SP-060056	0087	-	A	GBA keys handling and UICC presence detection	7.2.0	7.3.0	TEI
2006-03	SP-31	SP-060049	0089	-	A	Clarify the confusion of the useof NAF-ID and FQDN	7.2.0	7.3.0	(SEC1) (GAAext)
2006-03	SP-31	SP-060061	0090	-	F	key derivation clarifications	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060061	0091	-	F	Use of SIM for a Ua application	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-06	SP-32	SP-060424	0093	1	F	UICC removal	7.3.0	7.4.0	TEI
2006-06	SP-32	SP-060380	0095	-	A	Minimum lifetime of Keys	7.3.0	7.4.0	TEI7
2006-09	SP-33	SP-060500	0096	-	F	Clarification of notes	7.4.0	7.5.0	SEC7
2006-09	SP-33	SP-060500	0097	-	F	Corrections of some incorrect reference	7.4.0	7.5.0	GAA2
2006-09	SP-33	SP-060493	0099	-	A	Sending FQDN over Zn	7.4.0	7.5.0	SEC1-SC
2006-09	SP-33	SP-060496	0101	-	A	Correction to the UICC selection procedure in GBA	7.4.0	7.5.0	TEI6
2006-12	SP-34	SP-060810	0103	1	F	Addition of note on out-of-order case for BSF	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0104	1	F	Using pseudonyms over Zn to ensure service continuity	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0105	1	C	UICC application selection for service continuity	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0106	1	F	Two NAF applications having the same NAF keys	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060801	0110	1	F	Correction of Requirements on GBA keys	7.5.0	7.6.0	SEC7-GAA2
2006-12	SP-34	SP-060811	0111	-	C	Suppression of NAF Group attribute within USS on Zn and Zn' interfaces	7.5.0	7.6.0	SEC7-GAA2
2006-12	SP-34	SP-060809	0112	1	F	Addition of text regarding the use of the NDS authentication framework	7.5.0	7.6.0	NDSAFTLS
2007-03	SP-35	SP-070160	0113	1	F	Key deletion method for applications	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070160	0114	2	F	Encode problems in HTTP digest AKA authentication between UE and BSF	7.6.0	7.7.0	GAA2
2007-03	SP-35	SP-070160	0116	1	F	GBA key names clarification	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070160	0118	-	D	Correction of Note in Annex H	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070146	0120	1	A	Clarification on NAF_Id coding	7.6.0	7.7.0	TEI6
2007-03	SP-35	SP-070147	0122	-	A	Clarification of mapping of GUSS to IMPIs and IMSIs	7.6.0	7.7.0	SEC7-GAA2
2007-06	SP-36	SP-070338	0123	1	B	Details of HLR - BSF reference point	7.7.0	7.8.0	GAAExt
2007-06	SP-36	SP-070327	0124	-	F	Clarifying the terms 2G and 3G for GBA	7.7.0	7.8.0	2G GBA
2007-06	SP-36	SP-070327	0126	1	A	GBA NAF Keys storage policy in the UICC	7.7.0	7.8.0	SEC1-SC
2007-06	SP-36	SP-070340	0108	3	B	Introduction of temporary identifier for bootstrapping procedure	7.8.0	8.0.0	TEI8
2007-09	SP-37	SP-070594	0128	4	A	Correction to HLR - BSF reference point	8.0.0	8.1.0	SEC7-GAA2

2007-12	SP-38	SP-070792	0129	1	D	Addition of information for developers on B-TID uniquenss	8.1.0	8.2.0	TEI8
2007-12	SP-38	SP-070785	0132	1	A	2G GBA Certificate Management	8.1.0	8.2.0	TEI8
2007-12	SP-38	SP-070787	0134	1	A	Usage of OMA References – Update of Reference	8.1.0	8.2.0	TEI8
2008-03	SP-39	SP-080141	0138	1	A	Simultaneous handling of Zh' and Zh in a BSF	8.2.0	8.3.0	SEC7-GAA2
2008-03	SP-39	SP-080143	0136	-	F	Move Manual TLS certificate handling	8.2.0	8.3.0	TEI8
2008-09	SP-41	SP-080143	0139	1	F	Zh and Zh' intra-operator domain reference points	8.3.0	8.4.0	TEI8
2008-12	SP-42	SP-080744	0140	-	F	Add FC number space value allocations and clarification on length parameter	8.4.0	8.5.0	TEI8
2008-12	SP-42	SP-080744	0141	-	F	Using Unicode Standard Normalization Form when encoding using UTF-8	8.4.0	8.5.0	TEI8
2009-03	SP-43	SP-090137	0142	-	F	Add FC number space value allocations for HSPA SRVCC	8.5.0	8.6.0	TEI8
2009-06	SP-44	SP-090420	0142	1	F	Ua security protocol identifiers for IMS based MBMS	8.6.0	8.7.0	TEI8
2009-06	SP-44	SP-090420	0143	1	F	Clarify sending of MSISDN in Zn	8.6.0	8.7.0	TEI8
2009-06	SP-44	SP-090276	0142	-	C	Modification of References	8.7.0	9.0.0	TEI9
2009-06	SP-44	SP-090276	0144	1	B	GBA DIAMETER based Zn reference point to support TLS	8.7.0	9.0.0	TEI9
2009-09	SP-45	SP-090524	0143	-	F	FC value allocation for GPL	9.0.0	9.1.0	eGBAPush
2009-09	SP-45	SP-090524	0145	-	C	Introducing Ua security protocol Id for GPL	9.0.0	9.1.0	eGBAPush
2009-12	SP-46	SP-090820	0146	-	A	KDF clarification	9.1.0	9.2.0	eGBAPush
2009-12	SP-46	SP-090822	0148	-	F	Ua security protocol identifier for IMS media plane security	9.1.0	9.2.0	MEDIASEC
2010-06	SP-48	SP-100361	0149	1	F	Deprecation of SHA-1	9.2.0	9.3.0	TEI9
2010-10	SP-49	SP-100482	0150	1	C	Unification of TLS and certificate references in TS 33.220 with TS 33.310	9.3.0	10.0.0	TEI10
2011-09	SP-53	SP-110563	0151	-	F	Reintroduction of lost reference	10.0.0	11.0.0	Sec11
2011-12	SP-54	SP-110848	0153	1	F	Correction of Allowed TLS Ciphersuite Identifiers in Annex H.3	11.0.0	11.1.0	Sec11
2012-03	SP-55	SP-120033	0156	3	B	SIP Digest-based GBA scope and terminology updates	11.1.0	11.2.0	GBA-ext
			0159	2	B	GBA extension for re-use of SIP Digest credentials			
		SP-120039	0157	-	F	Update of TLS extensions version			Sec11
			0158	-	F	Correction of BSF and bootstrapping requirements in 2G GBA			
			0160	1	D	GBA terminology issues			
		SP-120037	0164	-	A	Correction of misimplementation of Change Requests on KDF FC value allocations			TEI8
2012-06	SP-56	SP-120340	0165	2	F	Introduction of auth-int in GBA Digest	11.2.0	11.3.0	GBA-ext
2012-06	SP-56	SP-120341	0166	-	F	Correction of TLS Extensions References to point toTS 33.310	11.2.0	11.3.0	SEC11
2012-06	SP-56	SP-120341	0167	-	F	Correction of phrase descr bing Zn procedure	11.2.0	11.3.0	SEC11
2012-06	SP-56	SP-120340	0168	-	F	NAF specific key derivation in GBA Digest	11.2.0	11.3.0	GBA-ext
2012-06	SP-56	SP-120340	0169	-	F	TMPI (temporary identity) support in GBA Digest	11.2.0	11.3.0	GBA-ext
2012-09	SP-57	SP-120605	0170	1	F	Correction of description of 'Bootstrapping Initiation'	11.3.0	11.4.0	SEC11
2013-03	SP-59	SP-130036	0171	1	F	Correction of references for GBA	11.4.0	12.0.0	SEC12, GBA-ext
2013-06	SP-60	SP-130249	0172	1	F	Ua security protocol identifier	12.0.0	12.1.0	Web_GBA
		SP-130255	0173	-	A	Removal of editor's note Release 12 - 33.220			FS_SSO_APS
		SP-130258	0175	1	F	Mandating encryption in the TLS profile for 2G GBA			TEI12, SEC7-2GGBA
		SP-130258	0176	1	F	Removal of realm check in 2G GBA			TEI12, SEC7-2GGBA
		SP-130258	0177	2	F	Correction of 2G GBA			TEI12, SEC7-2GGBA
2012-12	SP-62	SP-130667	0178	-	F	Checking that GBA types over Ua and Zn match	12.1.0	12.2.0	TEI12, GBA-ext
2014-06	SP-64	SP-140315	0180	1	C	Adding FC value for ProSe specification	12.2.0	12.3.0	ProSe

History

Document history		
V12.3.0	October 2014	Publication

EXHIBIT 1 TO FELDMAN REPLY DECLARATION

DOCUMENT 72

ONEM2M
TECHNICAL REPORT

Document Number	oneM2M-TR-0001-UseCase
Document Name:	oneM2M Use cases collection
Date:	2013-Sep-23
Abstract:	This oneM2M Technical Report includes a collection of use cases from various M2M industry segments. Use cases focus on the sequence of interactions among actors, and may include potential requirements.

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Copyright Notification

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1

1 Scope

The present document includes a collection of use cases from a variety of M2M industry segments (listed in table 1). Each use case may include a description, source, actors, pre-conditions, triggers, normal and alternative flow of sequence of interactions among actors and system, post-conditions, illustrations and potential requirements. The potential requirements provide an initial view of what oneM2M requirements could arise from the Use Case as seen by the contributor. These are intended to help the reader understand the use case's needs. These potential requirements may have been subsequently submitted by the contributor for consideration as candidate oneM2M requirements, which may or may not have been agreed as a oneM2M requirement (often after much editing). As such, there may not be a direct mapping from the potential requirements to agreed oneM2M requirements [i.15].

Table 1-1

Industry Segment	oneM2M Use Cases							
Agriculture								
Energy	Wide area Energy related measurement /control system for advanced transmission and distribution automa ion	Analytics for oneM2M	Smart Meter Reading	Environmental Monitoring for Hydro-Power Generation using Satellite M2M	Oil and Gas Pipeline Cellular /Satellite Gateway			
Enterprise	Smart building							
Finance								
Healthcare	M2M Healthcare Gateway	Wellness services	Secure remote pa ient care and monitoring					
Industrial								
Public Services	Street Light Automation	Devices, Virtual devices and Things	Car/Bicycle Sharing Services	Smart parking	Information Delivery service in the devastated area			
Residential	Home Energy Management	Home Energy Management System	Plug-In Electrical Charging Vehicles and power feed in home scenario	Real-time Audio/Video Communication	Event Triggered Task Execution	Semantic Home Control	Semantic Device Plug and Play	
Retail								
Transportation	Vehicle Diagnostic & Maintenance Report	Remote Maintenance services	Neighbourhood Alerting on Traffic Accident	Fleet management service using Digital Tachograph				
Other	Extending the M2M Access Network using Satellites	M2M data traffic management by underlying network operator	Optimizing connectivity management parameters with mobile networks	Optimizing mobility management parameters with mobile networks	Sleepy nodes	Collection of M2M system data	Leveraging Broadcasting/ Multicasting Capability of Underlying Networks	Service Provisioning for Equipment with Built-in Device

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

2.1 Normative references

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules (http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc)
[i.2] ETSI TR 102 935 v2.1.1, Machine to Machine communications (M2M);Applicability of M2M architecture to Smart Grid Networks; Impact of Smart Grids on M2M platform
[i.3] ETSI TS102 689 V1.1.1, Machine-to-Machine communications (M2M);M2M service requirements
[i.4] ETSI TR 102 732, Machine to Machine Communications (M2M); Use cases of M2M applications for eHealth
[i.5] ETSI TR 102 897, Machine to Machine Communications (M2M);Use cases of M2M applications for City Automation
[i.6] HGI-GD017-R3, Use Cases and Architecture for a Home Energy Management Service
[i.7] ISO/ IEC 15118 Road vehicles, vehicle to grid communication
[i.8] Mandate 486, MANDATE FOR PROGRAMMING AND STANDARDISATION ADDRESSED TO THE EUROPEAN STANDARDISATION BODIES IN THE FIELD OF URBAN RAIL
[i.9] DIN specification 70121, ELECTROMOBILITY - DIGITAL COMMUNICATION BETWEEN A D.C. EV CHARGING STATION AND AN ELECTRIC VEHICLE FOR CONTROL OF D.C. CHARGING IN THE COMBINED CHARGING SYSTEM
[i.10] ETSI TR 102 638, Intelligent Transport Systems (ITS);Vehicular Communications;Basic Set of Applications; Definitions
[i.11] 3GPP TS 22.386
[i.12] 3GPP TS 23.682, Architecture enhancements to facilitate communications with packet data networks and applications
[i.13] 3GPP TR 23.887, Architectural Enhancements for Machine Type and other mobile data applications
[i.14] Communications Guidelines defined in Continua Health Alliance, The Continua Version 2012 Design Guidelines
[i.15] oneM2M-TS-0002-Requirements Technical Specification
[i.16] ETSI TS103.383 Smart Cards; Embedded UICC; Requirements Specification
[i.17] IEC 61850 Communication networks and systems in substations

3 Acronyms

For the purposes of the present document, the following abbreviations apply:

A/C	Air Conditioner
AHD	Application Hosting Device
AL	Authorization Level
AMI	Advanced Metering Infrastructure
AMS	Asset Management System
AP	Applications Provider
API	Application Programming Interface
ARIB	Association of Radio Industries and Business
ARPU	Average Revenue Per User
BMS	Building Management System
BTS	Bus Ticket System
CCSA	China Communications Standards Association
CCTV	Closed Circuit Television
CIM	Common Information Model
CIP	Critical Infrastructure Protection
CIS	Customer Information System
CL	Criticality Level
CMS	Cryptographic Message Syntax
CP	Care Provider

CPU	Central Processing Unit
DER	Distributed Energy Resources
DMS	Distribution Management System
DNP	Distributed Network Protocol
DP	Device Provider
DR	Demand Response
DRX	Discontinuous reception
DSDR	Distribution Systems Demand Response
DSM	Demand Side Management
DSO	Distribution System Operator
DAP	Data Aggregation Point
DB	DataBase
DSRC	Dedicated Short Range Communications
DTG	Digital TachoGraph
DVR	Digital Video Recorder
ECU	Engine Control Unit
EGW	Energy GateWay
EHR	Electronics Health Record
EMS	Energy Management System
EPBA	Equipment Provider Back-end Application
ESB	Enterprise Service Bus
ESI	Energy Services Interface
ETRI	Electronics and Telecommunications Research Institute
ETWS	Earthquake and Tsunami Warning System
EV	Electric Vehicle
eUICC	Embedded Universal Integrated Circuit Card
EVC	Electric Vehicle Charging
EVCE	Electric Vehicle Charging Equipment
EVC-SP	Electric Vehicle Charging Service Provider
FAN	Field Area Network
FFS	For Further Study
FMS	Fleet Management Service
GPS	Global Positioning System
HAMS	Home Automation Management System
HAN	Home Area Network
HEM	Home Energy Management
HEMS	Home Energy Management System
HIPPA	Health Insurance Portability and Accountability Act
HMI	Human Machine Interface
HSM	Hardware Security Module
HV	High Voltage
ICCID	Integrated Circuit Card Identifier
IEC	International Electrotechnical Commission
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
ITS	Intelligent Transportation System
ITS-S	Intelligent Transportation System Station
KCA	Korean Communications Agency
KDDI	Kokusai Denshin Denwa International
LAN	Local Area Network
LATAM	Latin American
LDR	Low Data Rate
LG	Lucky Goldstar
MDMS	Meter Data Management System
MDM	Medical Device Manufacturer
MDN	Mobile Directory Number
MDMMS	Medical Device Monitoring & Management Service
MNO	Mobile Network Operator
MSCN	M2M Service Capabilities Network
MSISDN	Mobile Station International Subscriber Directory Number
MSP	M2M Service Platform
MTC	Machine Type Communications

MV	Medium Voltage
M2M	Machine to Machine
NAN	Neighborhood Area Network
NEC	Nippon Electric Company
NFC	Near Field Communications
NMS	Network Management System
NTT	Nippon Telegram and Telegraph
PAN	Personal Area Network
PC	Personal Computer
PEV	Plug-in Electric Vehicle
PHEV	Plug-In Hybrid Electric Vehicle
PKCS	Public Key Cryptology Standards
PLC	Power Line Communications
PMU	Phase Measurement Unit
QoS	Quality of Service
RL	Redaction Level
RSU	Road Side Unit
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SDDTE	Small Data and Device Triggering Enhancements
SDS	Samsung Data Systems
SGCG	Smart Grid Coordination Group
SGIP	Smart Grid Interoperability Panel
SIM	Subscriber Identity Module
SK	South Korea
SLA	Service Level Agreement
SM	Smart Meter
SMS	Short Message Service
SN	Sleepy Node
SP	Service Provider
SW	SoftWare
TNC	Trusted Network Connect
TPM	Trusted Platform Module
TPMS	Tire Pressure Monitoring System
TSO	Transmission System Operator
TTC	Telecommunications Technology Committee
TV	TeleVision
UD	User Device
UE	User Equipment
UEPCOP	User Equipment Power Consumption OPtimizations
UIM	User Identity Module
USB	Universal Serial Bus
VIP	Very Important Person
WAM	Wide Area Measurement
WAMS	Wide Area Measurement System
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
WG	Wireless Gateway
WLAN	Wireless Local Area Network
3GPP	3rd Generation Partnership Project

4 Conventions

The key words “Shall”, ”Shall not”, “May”, ”Need not”, “Should”, ”Should not” in this document are to be interpreted as described in the oneM2M Drafting Rules [i.1]

5 Energy Use Cases

5.1 Wide area Energy related measurement/control system for advanced transmission and distribution automation

5.1.1 Description

Background:

- Phase Measurement Units (PMUs, aka Synchrophasors) in power electrical systems , is a technology that provides a tool for power system operators and planners to measure the state of the electrical system and manage power quality.
- PMUs are positioned across the high voltage (HV) transmission and Medium voltage (MV) distribution network, operated by transmission and distribution system operators (TSO/DSO) respectively, typically in a substation where network node connections are made and the distribution of load is of importance.
- PMUs usually generate bulk statistical information transmitted hourly or daily or event based. They are capable of continuously monitoring the wide-area network status online, so continuous information streaming data will be available to control centres from hundreds of PMUs at once which requires a stable communication network with sufficient capacity and quality.
- The communications network that is used to collect, monitor and control electricity power systems (HV transmission and MV Distribution power systems) are usually owned by Electricity TSO/DSO and are very secure and reliable.
- PMUs are sampled from widely dispersed locations in the power system network and synchronized from the common time source of a global positioning system (GPS) radio clock. PMUs measure voltages and currents at diverse locations on a power grid and output accurately time-stamped voltage and current phasors, allowing for synchronized comparison of two quantities in real time. These comparisons can be used to assess system conditions.

Description:

- This use case shows the feasibility of High voltage /MV supervision through the interconnection of PMUs especially via mobile broadband communication networks. Thus not requiring any additional TSO/DSO internal network extensions especially in remote sites.
- Through analysis of PMU power state information collected in operator control centres (TSO/DSO), the TSO/DSO can send control information to PMUs, in the same mobile broadband communication network, to control the power flow in the power system.
- Transmission delay of less than a second for the transmission of PMU measurements in near real time to TSO/DSO in the case of control centres.
- Black-out causes propagates within minutes and sometimes only seconds through entire national and even international transport & distribution networks. So the transmission of control is critical in the range of less than seconds.

5.1.2 Source

Fujitsu, from ETSI TR 102 935 v2.1.1 [i.2]

5.1.3 Actors

- Energy system operators:

  - Transmission System Operator (TSO) is responsible for operation, maintenance and development of the transmission network in its own control area and at interconnections with other control areas, long-term power system ability to meet the demand, and grid connection of the transmission grid users, including the DSOs.
  - Distribution System Operator (DSO) is responsible for operation, maintenance and development of its own distribution grid and where applicable at the connections with other grids, ensuring the long-term ability to meet the distribution demand, regional grid access and grid stability, integration of renewables at the distribution level and regional load balancing (if that is not done by the balance responsible party).
- Communication operator (s) provider of the access network (Telcos)
  - System operators and/or providers of service layer platform(s) which can provide services/common functionalities for applications that are independent of the underlying network(s).

5.1.4 Pre-conditions

Communication/connectivity networks (phase network) to collect the measurements from PMUs to centers.

5.1.5 Triggers

System conditions deducted from the analysis of collected data trigger a counter measure action for example to curtail or reduce power flow in a HV/MV transmission.

5.1.6 Normal Flow

Interactions between actors and system required for successful execution of the use case or scenario.

An example flow for the TSO scenario:

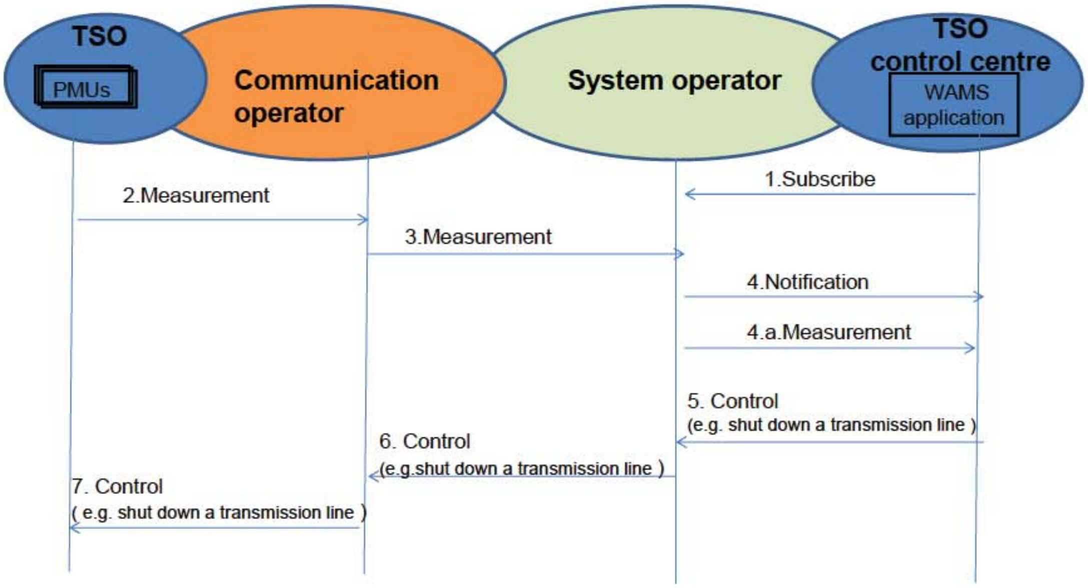


Figure 5-1 An example flow for the TSO scenario

1. WAMS application subscribes to PMU data which is owed by the Transmission System Operator
2. Measurements requested are sent back through (service provider) Telco operator and System Operator to TSO centre for the WAM application
3. Measurements sent to the system operator are collected and can be stored by the operator.
4. Notification message is sent to WAMS application in TSO control centre when the system operator receives the measurement. WAMS application/TSO control centre can pull/push the data measurements
5. Based on measurements collected, WAMS application/ TSO control centre initiates a control command to shut down a transmission line under its controlled area
6. The Control command is sent to system operator where an appropriate communication network is selected to send the control command
7. Then control command is sent by system operator to the PMU under TSO controlled area to initiate the execution of the command e.g. the shutdown of a specific transmission line

An example flow for DSO scenario:

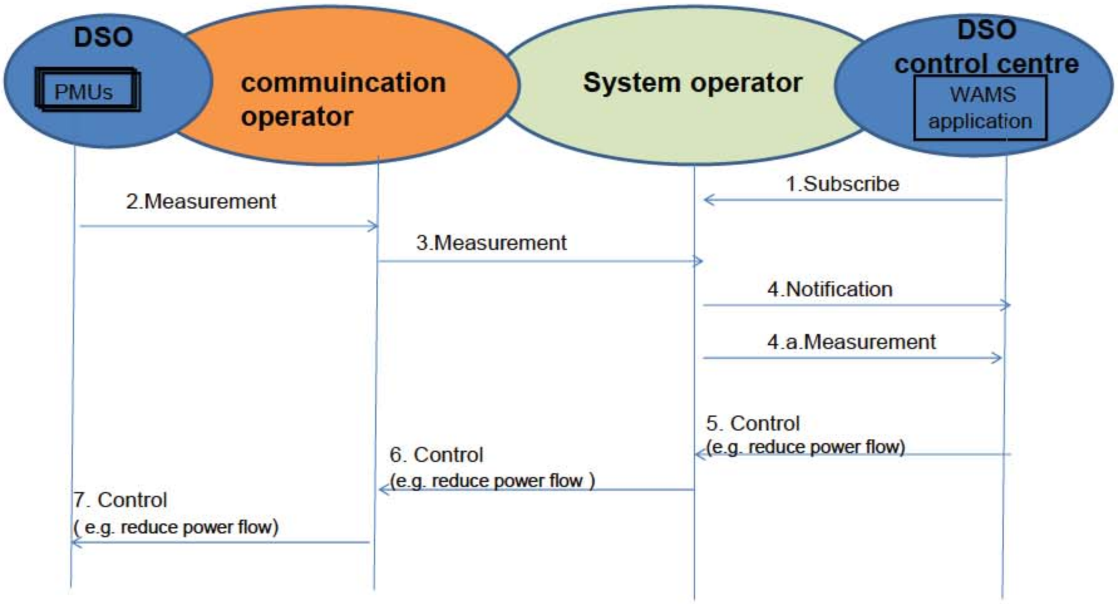


Figure 5-2 An example flow for DSO scenario

1. WAMS application subscribes to the PMU data

2. Measurements are sent through Telco operator

3. Measurements sent to system operator where they are stored.

4. Notification sent to WAMS application in DSO control centre when the measurements are received by system operator.WAMS application in DSO control centre pulls the measurements

5. Based on measurements collected WAMS application in DSO control centre, initiates a control command to reduce flow in a particular region under its controlled area.

6. Control command sent to system operator where an appropriate communication network is selected to send the control command.

7. Then control command is sent to the PMU under DSO control to initiate the execution of the command e.g. the change of power flow.

5.1.7 Alternative flow

None

5.1.8 Post-conditions

Corrective or Restricted operation of power electrical network as a result of the preventive action because of the shut-down of (a part) power network.

5.1.9 High Level Illustration

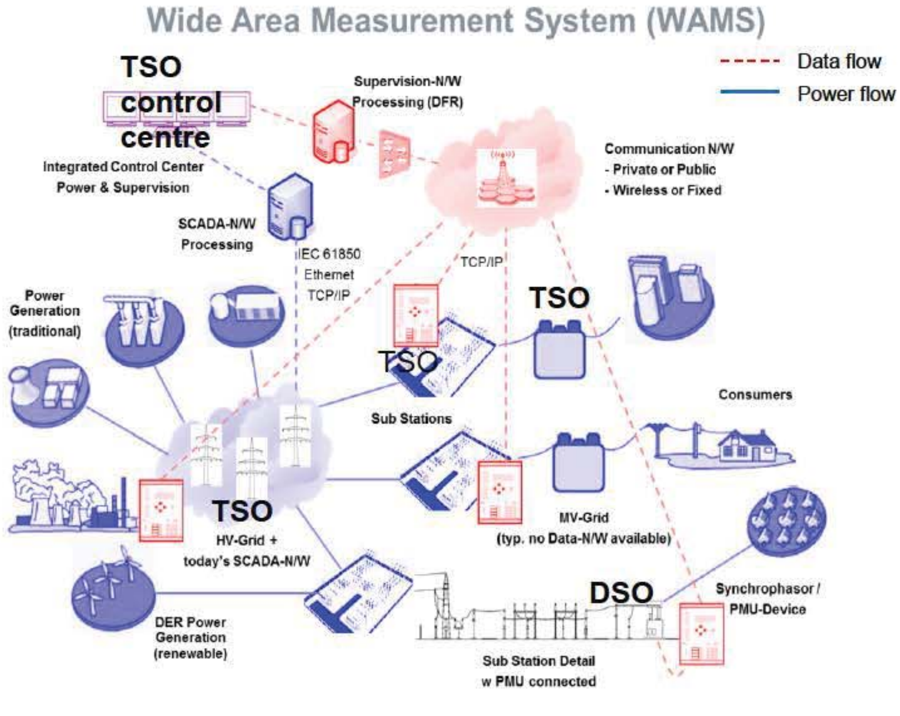


Figure 5-3 High Level Illustration of Wide Area Measurement System

5.1.10 Potential Requirements

Extracted from ETSI service requirements [i.3] (Ref TS102 689 V1.1.1) but suitable for this use case.

1. Data collection and reporting capability/function

   The M2M System (e.g. be owned by System Operator) shall support the reporting from a specific M2M Device (e.g. PMU) or group of M2M Devices or group of M2M collectors in the way requested by the M2M Application (e.g. WAM) as listed below:

   - a periodic reporting with the time period being defined by the M2M application;
   - an on-demand reporting with two possible modes. One is an instantaneous collecting and reporting of data, the other one is a reporting of the data that were pre-recorded at the indicated specific time period;
   - an event-based reporting e.g. transient fault (*Note specific time requirements FFS*)

2. Remote control of M2M Devices

   The M2M System shall support the capability for an Application to remotely control M2M Devices that support this capability; e.g. control power flow or shut down a regional power network to prevent a black-out event

3. Information collection & delivery to multiple applications

   The M2M System shall support the ability for multiple M2M Applications (in this use case the WAM) to interact with multiple applications on the same M2M Devices (in this case can interact with many PMUs) simultaneously

4. Data store and share

   The M2M System shall be able to store data to support the following requirements:

   - Provide functionality to store and retrieve data.
   - Establish storage policies for stored data (e.g. define maximum byte size of the stored data).
   - Enable data sharing of stored data subjected to access control

5. Security requirements

   a. Authentication of M2M system with M2M devices/ /collectors

      The M2M system shall support mutual authentication with M2M Device or M2M Gateway/collector. For example mutual authentication may be requested between a service providers/operators and the entity requesting the service. The parties may choose the strength of authentication to ensure appropriate level of security.

   b. Authentication of applications on M2M devices with M2M applications on the network

      When there is a request for data access or for M2M Device/Gateway access, the M2M Device or M2M Gateway access, the application on M2M Device or M2M Gateway shall be able to mutually authenticate or M2M Applications on the Network from which the access request is received.

   c. Data integrity

      The M2M System shall be able to support verification of the integrity of the data exchanged.

   d. Prevention of abuse of network connection

M2M security solution shall be able to prevent unauthorized use of the M2M Device/Gateway.

6. Privacy

   The M2M System shall be able to protect confidentiality of collected information.

   a. Security credential and software upgrade at the Application level.

      Where permitted by the security policy, M2M System shall be able to remotely provide the following features, at the Application level:

      - Secure updates of application security software and firmware of the M2M Device/Gateway.
      - Secure updates of application security context (security keys and algorithms) of the M2M Device/Gateway.

   b. This functionality should be provided by a tamper-resistant Secured Environment (which may be an independent Security Element) in M2M Devices/Gateways supporting this functionality.

7. Continuous Connectivity

   The M2M System shall support continuous connectivity, for M2M applications requesting the same M2M service on a regular and continuous basis. This continuous connectivity may be de-activated upon request of the Application or by an internal mechanism in the M2M system.

5.2 Analytics Use Case for M2M

5.2.1 Description

The term “analytics” is often used to describe complex algorithms applied to data which provide actionable insights. Simpler algorithms may also provide actionable insights – here we use the term “compute” for them. Both “analytics” and “compute” may be used similarly by an M2M System to provide benefits to M2M applications. This use case uses a simple “compute” example to introduce the topic.

M2M application service providers may wish to use analytics for several purposes. There are many analytics providers who may offer their libraries directly to application service providers. However there are situations where application service providers may wish to apply analytics to their M2M data from devices before it is delivered to the “back-end” of the application “in the cloud”.

To satisfy M2M application service provider needs, a oneM2M system may offer compute/analytics capabilities which may be internally or externally developed. Furthermore, these compute/analytics capabilities may be geographically distributed. Benefits to M2M application service providers might include:

- Convenience - due to integration
- Simplicity - due to a cross-vertical standardized analytics interface
- Cost savings – due to resource minimization (of compute, storage, and/or network)
- Improved performance – due to offloading/edge computing

M2M service providers may also benefit by deploying distributed compute/analytics to optimize operations such as regional management e.g. device/gateway software updates.

The use case described below assumes:

- millions of devices continuously report M2M data from devices at geographically diverse locations
- the M2M application is interested in receiving only certain sets of data based upon changes in particular data elements.

Use of oneM2M computation and analytics for anomaly detection and filtering avoids the use of bandwidth needed to transport unnecessary device data to the back-end of the M2M application. To enable the oneM2M system to do this, the M2M application specifies:

1. Which device data (the baseline set) is needed to create a baseline (which is indicative of “normal” operation).
2. The duration of the training period used to set a baseline
3. The method to create/update the baseline
4. Which device data (the trigger set) is to be compared to the baseline
5. The method of comparison between the baseline set and the trigger set.
6. The variation of M2M data in comparison to the baseline used to trigger action
7. Which data (the storage set) is to be stored in addition to the data used in the baseline.
8. Which data (the report set, which may include data from the baseline set, trigger set and the storage set) which is to be reported to the M2M application upon trigger.

This is a draft oneM2M document and should not be relied upon; the final version, if any, will be made available by oneM2M Partners Type 1

9. “Location directives” which expresses where the device data collection point, storage and compute/analytics program and libraries should be located. (Distributed, possibly hierarchical locations may be specified, and may be defined by max response time to devices, geographic location, density of convergent device data flows, available compute/storage capacity, etc.).

10. “Lifecycle management directives” for compute/analytics program and libraries instances e.g. on virtual machines.

The action by the oneM2M system in response to a trigger in this use case is to send the filtered report set to the M2M application; however, other alternative actions are summarized below (which would require different information from the M2M application).

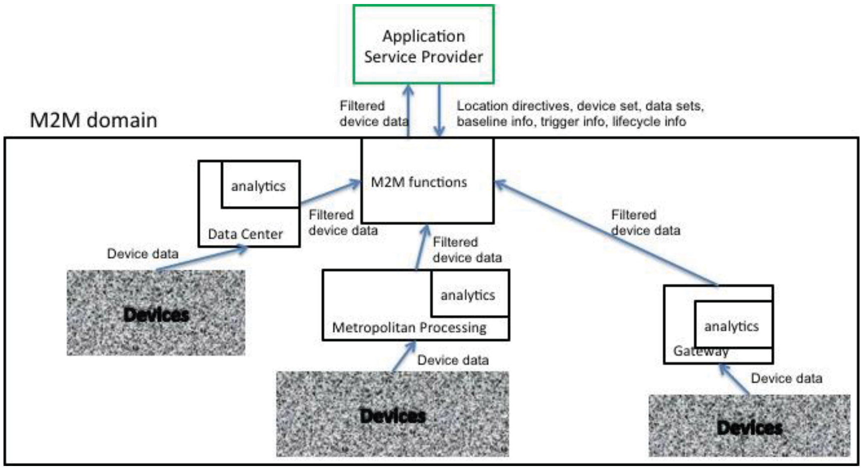


Figure 5-4 Analytics Use Case for M2M

Example of distributed, non-hierarchical location of analytics use case – normal flow

A hierarchical version of this use case would locate different compute/analytics at different levels of a hierarchy.

5.2.2 Source

Cisco Systems

5.2.3 Actors

Devices – aim is to report what they sense

Analytics library provider – aim is to provide analytics libraries to customers

M2M application service provider – aim is to provide an M2M application to users

5.2.4 Pre-conditions

Before an M2M system's compute/analytics may be used, the following steps are to be taken:

1. The M2M application service provider requests compute/analytics services from the oneM2M system. A request may include parameters required by analytics to perform computation and reporting, plus parameters required by the oneM2M system to locate and manage the lifecycle of the analytics computation instance (see 5.2.1).

2. The oneM2M system selects a source Analytics library provider for, and obtains the appropriate analytics library.

3. The oneM2M system provisions the appropriate analytics library at a location that meets the M2M application service provider's location directives.

4. The oneM2M system generates a program based upon the M2M application service provider's request.

5. The oneM2M system provisions the appropriate program based upon the M2M application service provider's request at the location(s) of step 3.

6. The oneM2M system starts collecting M2M data from devices and inputs them into the provisioned compute/analytics program for the duration of the baseline-training period. A baseline is established, which may include bounds for M2M data ranges, bounds for frequency of M2M data received, bounds for relative M2M data values to other M2M data values, etc.

5.2.5 Triggers

Triggering is described within 5.2.7.

5.2.6 Normal Flow

7. The devices provide M2M data to the oneM2M system.

8. The oneM2M system stores a set of M2M data (the storage set) from the devices

9. The oneM2M system uses analytics to compare M2M data (the trigger set) from devices with the baseline.

10. The oneM2M system determines whether the variation between the M2M data set and the baseline exceeds the specified bounds of the trigger condition, if it does then the following action occurs:

11. The oneM2M system sends the requested M2M data (the report set), to the M2M application service provider.

5.2.7 Alternative Flow 1

The action to be taken by the oneM2M system following a trigger may be different than step 11 above.

For example, the action may be to initiate conditional collection where for some duration or until some other trigger occurs.

A. A current collection scheme of device data is modified e.g. more frequent updates, or

B. A new collection scheme is initiated

Other alternative actions may include, but are not limited to:

- Initiating device/gateway diagnostics e.g. following a drop in the number of responding devices
- Sending control commands to devices
- Sending alerts to other oneM2M system services e.g. fraud detection
- Sending processed (e.g. cleansed, normalized, augmented) data to the application

5.2.8 Post-conditions

None.

5.2.9 High Level Illustration

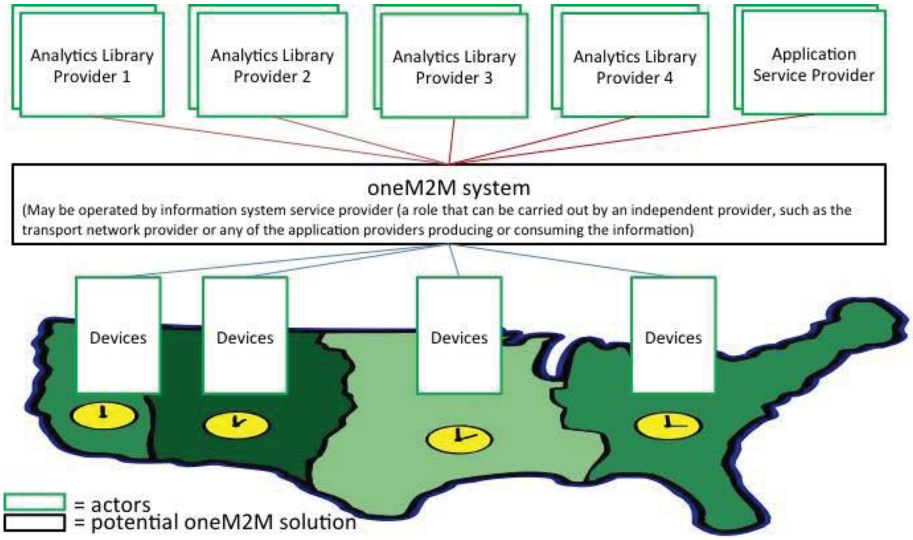

Analytics Library Provider 1
Analytics Library Provider 2
Analytics Library Provider 3
Analytics Library Provider 4
Application Service Provider
oneM2M system
(May be operated by information system service provider (a role that can be carried out by an independent provider, such as the transport network provider or any of the application providers producing or consuming the information)
Devices
Devices
Devices
Devices
= actors
= potential oneM2M solution

Figure 5-5 High level illustration of Analytics use case

5.2.9.1 Concrete Example Oil and Gas

The above description is of the abstracted use case; a more concrete example is as follows:

Oil and gas exploration, development, and production are important potential use cases for M2M. To stay competitive energy companies are continuously increasing the amount of data they collect from their field assets, and the sophistication of the processing they perform on that data. This data can literally originate anywhere on Earth, is transported to decision makers over limited bandwidths, and often must be reacted to on real-time time scales. An M2M system can prove very useful in its ability to perform analytics, data storage, and business intelligence tasks closer to the source of the data.

Oil and Gas companies employ some of the most sophisticated and largest deployments of sensors and actuators networks of any vertical market segment. These networks are highly distributed geographically, often spanning full continents and including thousands of miles of piping and networking links. Many of these deployments (especially during the exploration phases) must reach very remote areas (hundreds of miles away from the nearest high bandwidth Internet connection), yet provide the bandwidth, latency and reliability required by the applications. These networks are typically mission critical, and sometimes life critical, so robustness, security, and reliability are key to their architecture.

Oil and gas deployments involve a complex large-scale system of interacting subsystems. The associated networks are responsible for the monitoring and automatic control of highly critical resources. The economic and environmental consequences of events like well blowouts, pipeline ruptures, and spills into sensitive ecosystems are very severe, and multiple layers of systems continuously monitor the plant to drive their probability of occurrence toward zero. If any anomalies are detected, the system must react instantly to correct the problem, or quickly bring the network into a global safe state. The anomalies could be attributable to many different causes, including equipment failure, overloads, mismanagement, sabotage, etc. When an anomaly is detected, the network must react on very fast timescales, probably requiring semi-autonomous techniques and local computational resources. Local actions like stopping production, closing valves, etc. often ripple quickly through the entire system (the system can’t just close a valve without coordinating with upstream and downstream systems to adjust flows and insure all parameters stay within prescribed limits). Sophisticated analytics at multiple levels aids the system in making these quick decisions, taking into account local conditions, the global state of the network, and historical trends mined from archival big data. They may help detect early signs of wear and malfunction before catastrophic events happen.

Security is critical to Oil and Gas networks. This includes data security to insure all data used to control and monitor the network is authentic, private, and reaches its intended destination. Physical security of installations like wells, pump stations, refineries, pipelines, and terminals is also important, as these could be threatened by saboteurs and terrorists.

There are three broad phases to the Oil and Gas use case: Exploration, Drilling and Production. Information is collected in the field by sensors, may be processed locally and used to control actuators, and is eventually transported via the global internet to a headquarters for detailed analysis.

Exploration

During the exploration phase, where new fields are being discovered or surveyed, distributed process techniques are invaluable to manage the vast quantities of data the survey crews generate, often in remote locations not serviced by high bandwidth internet backbones. A single seismic survey dataset can exceed one Petabyte in size. Backhauling this data to headquarters over the limited communications resources available in remote areas is prohibitive (Transporting a petabyte over a 20Mb/s satellite link takes over 12 years), so physical transport of storage media is currently used, adding many days of time lag to the exploration process. Distributed computing can improve this situation. A compute node in the field is connected to the various sensors and other field equipment used by the exploration geologists to collect the data. This node includes local storage arrays, and powerful processor infrastructures to perform data compression, analysis, and analytics on the data set, greatly reducing its size, and highlighting the most promising elements in the set to be backhauled. This reduced data set is then moved to headquarters over limited bandwidth connections.

Drilling

When oil and gas fields are being developed, large quantities of data are generated by the drilling rigs and offshore platforms. Tens of thousands of sensors monitor and record all conditions on the rig, and thousands of additional sensors can be located downhole on the drill string, producing terabyte data sets. Distributed compute nodes can unify all of these sensor systems, perform advanced real-time analytics on the data, and relay the appropriate subset of the data over the field network to headquarters. Reliably collecting, storing and transporting this data is essential, as the future performance of a well can be greatly influenced by the data collected and the decisions made as it is being drilled.

A subset of the data collected (wellhead pressure, for example) is safety critical, and must be continuously analyzed for anomalies in real-time to insure the safety of the drilling operations. Because of the critical latency requirements of these operations, they are not practical for the Cloud, and distributed computing techniques are valuable to achieve the necessary performance.

Production

Once wells are producing, careful monitoring and control is essential to maximize the productivity of a field. A field office may control and monitor a number of wells. A computing node at that office receives real-time reports from all the monitoring sensors distributed across the field, and makes real-time decisions on how to best adjust the production of each well. Some fields also include injection wells, and the computing node closes the feedback loop between the injection rates and the recovery rates to optimize production. Some analytics are performed in the local computing node, and all the parameters are stored locally and uplinked to headquarters for more detailed analysis and archiving. Anomalies in sensor readings are instantly detected, and appropriate reactions are quickly computed and relayed to the appropriate actuators.

The Pump Station shown also includes a computing node. It is responsible for monitoring and controlling the pumps / compressors responsible for moving the product from the production field to the refinery or terminal in a safe and efficient manner. Many sensors monitor the conditions of the pipelines, flows, pressures, and security of the installation for anomalous conditions, and these are all processed by the local computing node.

Conclusion

The oneM2M Services Layer could offer “cloud-like” services to M2M Applications of computation/analytics functions commonly used across verticals, where those functions are optimally placed near to the sources of M2M data.

These services could include:

1. Advertisement of services to M2M Applications
2. Acceptance of M2M Applications’ directives over the “North-bound” interface.
3. Selection of where the requested computation/analytics functions are optimally placed
4. Provisioning and maintenance of virtual machine and computation/analytics functions (provided by oneM2M provider or 3rd party)
5. Redirection of M2M traffic to the virtual machine
6. Delivery of virtual machine output to other virtual machines or directly to M2M Applications (e.g. of filtered M2M data)

The M2M Applications and the M2M Service Provide may benefit from these services:

oneM2M Services Layer use of virtual machines on behalf of M2M Applications (e.g. to trigger new/modified data collection or device diagnostics or low latency M2M Device control)

oneM2M Services Layer use of virtual machines on behalf of the oneM2M Service Provider (e.g. optimized device management, fraud detection)

5.2.10 Potential requirements

1. The oneM2M system should be able to accept standardised inputs from M2M application providers which request compute/analytics services.

   Note: Many Analytics APIs exist today, the most popular one being Google analytics service

2. The oneM2M system should be able to select analytics libraries from Analytics library providers.
3. The oneM2M system should be able to locate and run instances of compute/analytics programs and libraries at locations requested by M2M applications service providers.
4. The oneM2M system should be able to manage the lifecycle of instances of compute/analytics programs and libraries.
5. The oneM2M system should be able to steer device data to inputs of instances of compute/analytics programs
6. The oneM2M system should be able to take operational and management action as a result of analytics reports received.
7. The oneM2M system should specify supported compute/analytics triggers and actions.

5.3 Smart Meter Reading

5.3.1 Description

This clause provides selected Smart Meter Reading use cases

5.3.2 Source

Qualcomm (contributor), use case information extracted from SGIP/OpenSG

5.3.3 Actors

Smart Meters (SM), Data Aggregation Points (DAPs), Advanced Metering Infrastructure (AMI) Head-end, Meter Data Management System (MDMS), Customer Information System (CIS)

5.3.4 Pre-conditions

Availability of meter data

5.3.5 Triggers

Smart meter on-demand or bulk interval meter read request events

5.3.6 Normal Flow

Smart Grid Interoperability Panel (SGIP)(http://www.sgip.org) and OpenSG users group (http://osgug.ucaiug.org/default.aspx) have been leading this effort in North America. An informative document has been submitted to OneM2M based on the SGIP activity. In general, a number of external organizations such as the SGIP or the SGCG (Smart Grid Coordination Group) in Europe have been working to define use cases for Smart Grid (SG). Portals such as the Smart Grid Information Clearing House (http://www.sgiclearinghouse.org) to assist with distributing information about smart grid initiatives in the US. The use-cases presented are derived in part from the above publicly available information.

Figure 5-6 shows the conceptual actors/data flow diagram based on a more detailed diagram developed by SG-Net. The more detailed diagram developed by SG-Net can be seen in the associated submission related to SGIP-based Smart Grid Use Cases.

In Figure 5-7 each element is an “actor” that is communicating with another actor using the shown data flows. As an example, consider “Smart Meter” in the “Customer” quadrant (lower right). Smart Meter (SM) communicates with a number of other actors, such as a Data Aggregation Point (DAP) located in the AMI Network. The DAP can then transmit the aggregated data to the Utility Service Provider using the Wide Area Network. The meter reading information can reach the data center for the Utility Service Provider via the AMI Headend which can forward the information to the MDMS which can coordinate with the CIS to store/retrieve meter data and to determine customer billing information. In certain variations such as cellular-based smart metering systems, a DAP entity may be bypassed, or merely serve as a pass-through for the information flow between the utility data center and the smart meter.

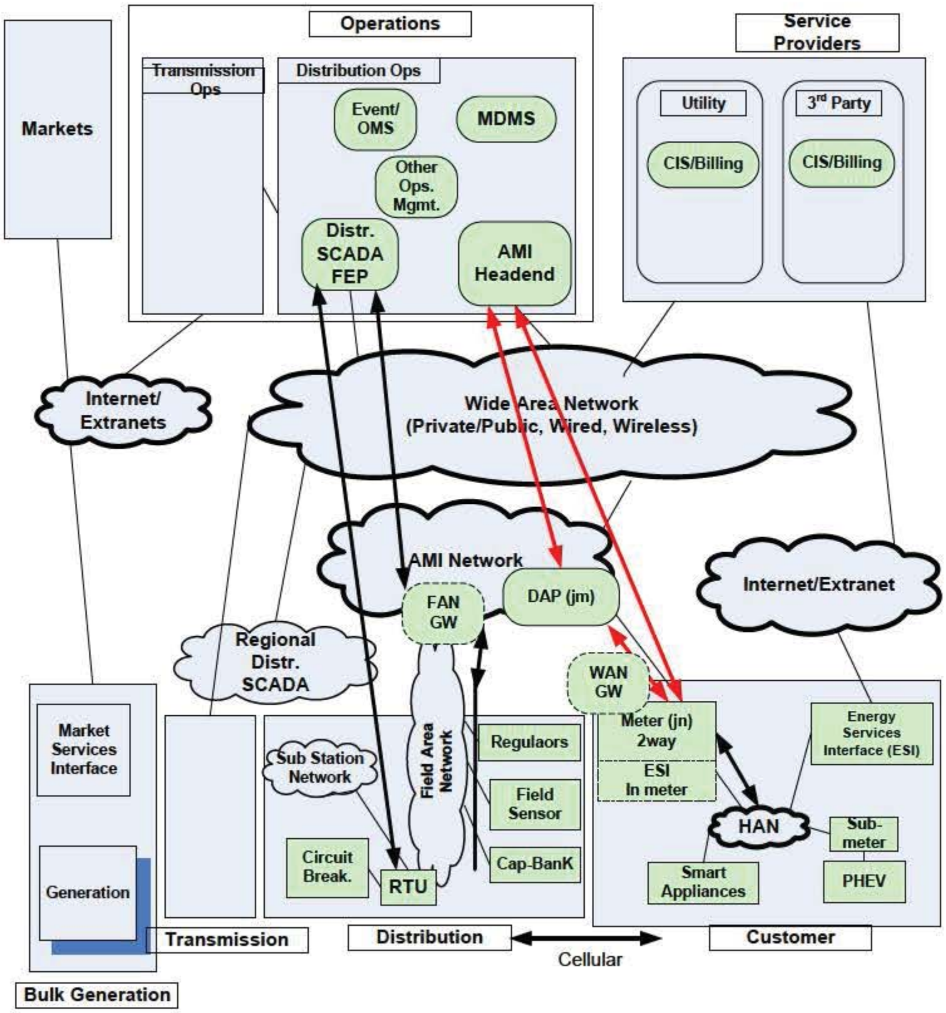


Figure 5-6 Conceptual Actors/Data Flow Diagram

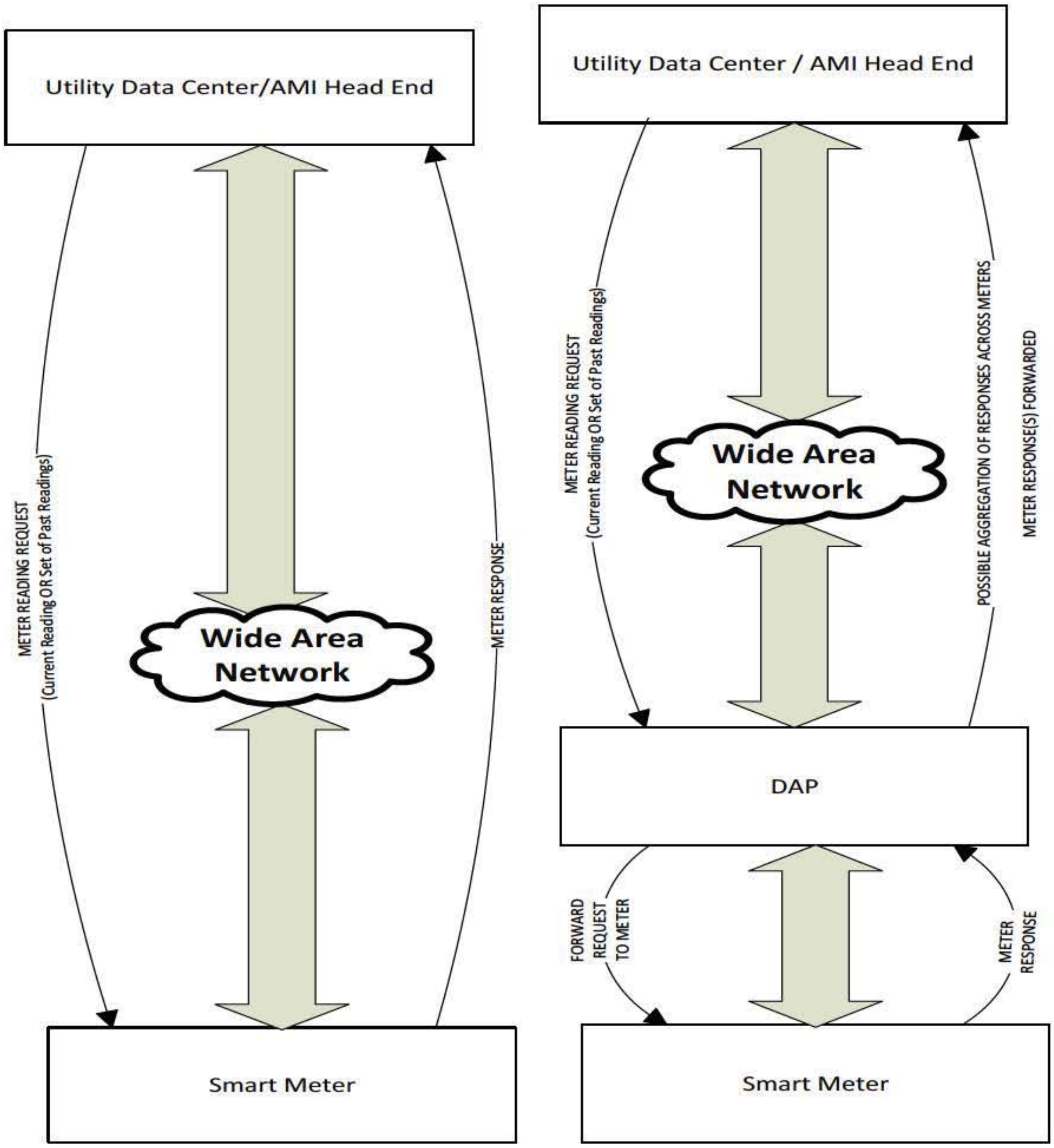

Utility Data Center/AMI Head End
Wide Area Network
Smart Meter
METER READING REQUEST (Current Reading OR Set of Past Readings)
METER RESPONSE
Utility Data Center / AMI Head End
Wide Area Network
DAP
Smart Meter
METER READING REQUEST (Current Reading OR Set of Past Readings)
POSSIBLE AGGREGATION OF RESPONSES ACROSS METERS
METER RESPONSE(S) FORWARDED
FORWARD REQUEST TO METER
METER RESPONSE

Figure 5-7 Typical Smart Meter Reading Flows A (on left) and B (on right)

Typically, a utility data center processing application communicates end-to-end via the AMI Headend with a smart meter data application at the edge. Figure 5.3.6-2 shows two possible flows A and B depending on whether there is a DAP entity along the path from the Utility Data Center / AMI Headend and the Smart Meter.

In flow A, the Utility Data Center / AMI Headend can make a request to the Smart Meter directly. Typically there may be 3 to 6 such requests per day (typically < 10 times per day). The request could indicate that the current meter reading is desired. Alternatively, multiple meter readings over a period of time such as for a few hours (e.g. from 2 p.m. to 8 p.m.) for a given day or across days could be requested. The Smart Meter completes the request and communicates it back to the Utility Data Center / AMI HeadEnd. Typical in such on-demand or bulk-interval read requests, a reasonably immediate response is desired of the order of a few seconds, so that there is not necessarily any significant delay tolerance allowed for the response. However, it is possible that, in current systems or in future systems, such requests could optionally carry a delay tolerance associated with the request depending on the urgency of the request. The size of the meter reading response can be of the order of a few tens to hundreds of bytes, and is also implementation dependent.

In flow B, the Utility Data Center / AMI Headend can make a request to the Smart Meter that can be received via the DAP. Typically there may be 3 to 6 such requests per day (typically < 10 times per day). The request could indicate that the current meter reading is desired or that multiple meter readings over a period of time are desired. The Smart Meter completes the request and sends its response to the DAP. This response from the Smart Meter to the DAP is typically desired in the order of 15 to 30 seconds, as suggested in the submitted informative document related to SGIP-based Smart Grid Use Cases. However the actual delay in processing can be implementation dependent across smart metering systems across the world. The size of the meter reading response can be of the order of a few tens to hundreds of bytes, and is also implementation dependent. The DAP entity can subsequently buffer the data for some time, receive data from many meters, and then submit the aggregated data across meters to the Utility Data Center / AMI Head End. The duration for which the DAP may buffer data can be implementation dependent, and could last for several seconds or minutes. In some variants, the DAP may serve merely as a router, so that it directly forwards the smart meter response to the Utility Data Center / AMI HeadEnd without performing any aggregation tasks. In further variants, the DAP entity could be merely a virtual processing entity and not a physical one, where such a virtual entity could even potentially reside on the other side (not shown) of the wide area network associated with the Utility Data Center / AMI Head End.

Summary

To summarize, meter reading requests could request a single meter reading or a set of meter readings. Such requests may occur a few times (typically < 10) per day and can be of the order of a few tens of bytes. Meter reading responses can be of the order of a few 10s to 100s of bytes typically. Meter reading responses are typically expected in the order of a few seconds after reception of the request at the meter. Any delay tolerance associated with such requests can be optional or implementation dependent. In some system variants, a DAP entity may not exist at all so that the Utility Data Center / AMI Head End communicates directly with the smart meter. In other end-to-end system variants, a DAP entity may serve as an intermediate processing or forwarding entity between the Smart Meter and the Utility Data Center / AMI Head End. In such cases, the DAP entity may be either a physical or virtual processing entity in the end-to-end system and can assist with buffering and aggregating meter reading responses. The duration of buffering or aggregation at the DAP entity can be implementation dependent and could be of the order of a few seconds or minutes typically.

5.3.7 Alternative flow

None

5.3.8 Post-conditions

None

5.3.9 High Level Illustration

None

5.3.10 Potential Requirements

None

5.4 Environmental Monitoring of Remote Locations to Determine Hydropower

5.4.1 Description

Monitoring environmental parameters and effects in remote locations is of increasing interest due to the rapidly changing Global Climate and the world in general. Parameters such as temperate, pressure, water levels, snow levels, seismic activity have significant effects on applications such as green energy (wind and hydro power), agriculture, weather forecasting and tsunami warnings. The demand for remote monitoring information (real time and historical) has been increasing over the past decade and expected to increase exponentially in the foreseeable future.

Environmental monitoring is a M2M application where satellite is the only communications alternative as no other infrastructure is generally in such remote localities. This case study attached presents one solutions where satellite communication is commonly used for environmental monitoring. This is Hydro power generation through snow/water monitoring.

This attached paper provides an overview of the solution and how satellite is used to support this requirement. The document also outlines why the solution requires M2M remote satellite communications.

5.4.2 Source

Inmarsat

5.4.3 Actors

Energy companies

5.4.4 Pre-conditions

Two main requirements exist for remote monitoring in Hydro Power Generation. Firstly, there needs to be monitoring of the flow and supply of water to generate the power itself. Secondly, there needs to be monitoring of the environmental impact the hydro-electricity has on surrounding ecosystems for the storage of water and resulting change in natural flow.

Flow and Supply of Water: Availability and supply of water is fundamental to hydro generated power and is very seasonal and related to the regional climate. In cold climates such as Canada and Norway, water is supplied by snow where reservoirs are located in high locations and catchment areas cover extensive mountain regions. Snow levels, melting periods and supplies are inconsistent throughout the year. Reservoirs and storage facilities are designed to take into account seasonal inconsistencies from mother nature. In more tropical areas such as Brazil, tropical downfalls in the wet seasonal periods are important for flow management and are also seasonal.

Regardless of region, accurate sensors are critical to monitor water flow and supply such as rain fall, snow levels, snow temperature, snow wetness, reservoirs levels and other seasonal parameters. These sensor readings are critical to ensure Hydro companies can accurately predicate and monitor power generation levels. Sensor readings need to be sent back in near real time to Hydro processing plants to maintain operations. The location for the sensors are in mountainous and hard to reach areas, that experience harsh environmental factors, partially high water/snow falls. Power or communication infrastructure is generally not available; therefore reliable satellite communication is the only option.

Sensor data is sent back consistently at short interval rates generally every five minutes from a number of multiple sensors in each location. Monthly usages in the region of 5 MB-10MB per month are typical depending on the number of sensor registers to poll and the M2M SCADA (supervisory control and data acquisition) communication protocol used (e.g. Modbus or priority protocol protocols used such as Totalflow).

Environmental impact that hydro-electricity has on surrounding ecosystems: Hydro-Electricity has the potential to affect the local ecosystems upstream and downstream from the generating plants. Government and world regulations are in place to ensure these systems minimise the impact on the local environment. Close monitoring and reporting of the surrounding areas are also part of the monitoring solution. Factors such as soil salinity, water levels, fish stock levels and erosion are some parameters that could be potentially monitored to ensure regulation and adhered to. This type of data is not critical for the power generation, however is required historically for trend analysis. Near real time communications is require for these types of sensors.

Sensor data is sent back long consistently interval rates generally every 30 minutes to 1 hour from a number of multiple sensors in each location. Monthly usages in the region of 1 MB-2 MB per month are typical, depending on the number of sensor registers to poll and the M2M SCADA communication protocol used.

5.4.5 Triggers

Two triggers that initiate information being sent over this architecture.

- Constant polling and
- Conditional polling.

Constant Polling: Sensor polling rates are set by the Hydro operator. This information is used at the host to provide real time data as well as historical for trending analysis. Polling rates depend on the rate of change in environmental changes or how often data is required to make decision on flow rates through the Pembroke. Rates could be every few minutes up to few hours, but rates are constant. This data is very important to determine power requirements for the satellite terminal. The more data the more power that is required.

Conditional Polling: Information can be sent from the RTU based on specified events, sharp rise in water levels, temperate and any specific data. This data must be fed back to the Hydro control (host) in the event critical controls need to be made on the Hydro station.

5.4.6 Normal Flow

Remote Sensor/Satellite Terminal Integration: Remote sensors are normally connected to a Remote Terminal Unit (RTUs) that condition the sensors values into registers that are transmitted (over satellite) to a host. The RTU polls (or changes register value in some circumstances) register values from Programmable Logic Controllers (PLCs) that are connected to the aforementioned sensors. The RTU will then use a M2M (SCADA) communication protocol to send the register values to the host. SCADA protocol are designed to be very compact, only sending the minimum require data to the host, thus why serial based communication is popular. Modbus, DNP3 (Distributed Network Protocol), IEC 61850 [i.17] (used in electrical substations) or other priority based communication protocols are used and are generally based around serial communication to keep traffic to a minimum. IP is starting to become more popular to support these SCADA protocols.

The host resides in a corporate network of the Hydro provider, which analyses and presents this data into meaning information to make decisions on. The host is normally a hydro-power monitoring application designed specifically by the hydro provider that is integrated with the remote monitoring sites and controls for the Hydro plant. The host normally has a very advanced Human Machine Interface (HMI) to process data to a human operator, and through this, the human operator monitors water flow and controls the amount of water flowing through the penstock to the turbine.

As mentioned, RTUs communicate via either serial (RS-232/485) or IP layer 2 M2M SCADA protocols. Majority of modern based satellite communications systems support IP only layer two protocols and it is very common for RTUs to communicate via serial only. Terminals servers are usually placed in line between RTUs and satellite terminals where serial communication is required.

Satellite Service solution: L Band satellite service are the most popular used by Hydro plants in LATAM and North America. The L band satellite service operates over the L band frequency range (1.5GHz to 1.6GHz). This band is unique as it is not attenuated by weather where other high frequency band solutions operate in. Remote terminals in this application must be able to operate in wet tropical and cold snow ranges.

The terminal normally provides a direct IP network connection to the customer corporate control network (backhaul) via secure IP VPNs or leased line. A backhaul satellite solution is sometimes used for increase reliability. The L band satellite network must offers geographical redundancy for downlink earth station and backhaul infrastructure.

Satellite Terminal Solution: The L band satellite terminal must operate with extremely low power, less than 1W idle and 20W transmit. Majority of power used by remote terminals is used during the idle state. Solar power designs are suitable for the most modern L band satellite terminals terminal to operate in remote locations.

Remote terminal management and control is essential for this remote application. The terminal must continually ensure the terminal is on-net. If the terminal seems to be unable to transmit (or receive), the terminal automatically must reboots and reconnects itself to the network (known as watchdog). This removes the requirement to send someone to reboot the terminal. Remote management is conducted via out of band signaling. Terminal status, manual reboot and remote firmware updates are also essential of the operation of the remote terminal.

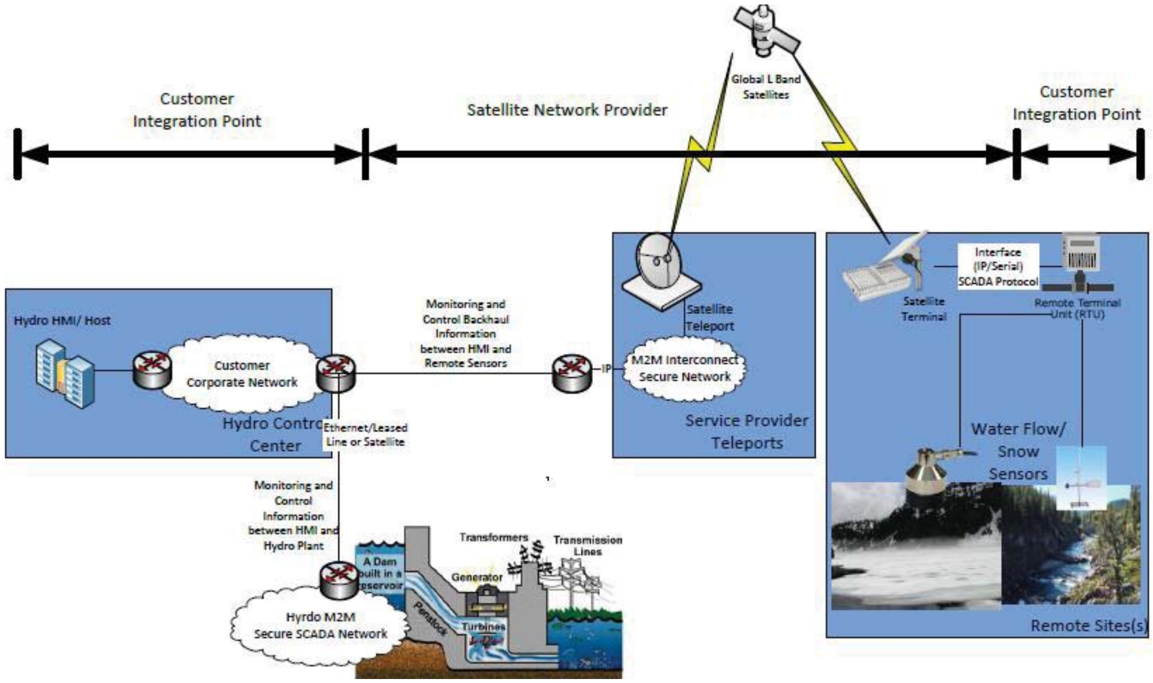
5.4.7 Alternative flow

None

5.4.8 Post-conditions

None

5.4.9 High Level Illustration



Global L Band Satellites
Customer Integration Point
Satellite Network Provider
Customer Integration Point
Hydro HMI/ Host
Customer Corporate Network
Hydro Control Center
Ethernet/Leased Line or Satellite
Monitoring and Control Backhaul Information between HMI and Remote Sensors
IP
Satellite Teleport
M2M Interconnect Secure Network
Service Provider Teleports
Satellite Terminal
Interface (IP/Serial) SCADA Protocol
Remote Terminal Unit (RTU)
Water Flow/ Snow Sensors
Remote Sites(s)
Monitoring and Control Information between HMI and Hydro Plant
A Dam built in a reservoir
Transformers
Transmission Lines
Generator
Penstock
Turbines
Hyrdo M2M Secure SCADA Network

Figure 5-8 High Level Illustration of Environmental Monitoring for Hydro-Power Generation using Satellite M2M

5.4.10 Potential Requirements

1. The M2M System shall provide mechanisms for ensuring round trip communications of specified times from sensors to actuators.

2. The M2M System shall support power constrained devices.

3. The M2M System shall support an M2M Application’s choice of communications transport characteristics e.g. Reliable or unreliable.

4. The M2M System shall support commonly used communications mechanisms for local area devices, e.g. RS-232/RS422.

5. The M2M System must provide communication availability to exceed 99.5% (1.83 days/year).

5.5 Oil and Gas Pipeline Cellular/Satellite Gateway

5.5.1 Description

This use case addresses a cellular gateway to transport oil and gas pipeline data to a backend server, to remotely monitor, manage and control devices equipped in the pipeline (e.g. meters, valves, etc.).

Oil and gas companies can have meters are remote destinations that makes manual monitoring of the state of these meters as an expensive task to be pursued on a regular basis. Automated monitoring of oil and gas pipeline data can streamline the remote monitoring and management of these remote pipeline meters.

When a fault is monitored on specific link of the pipeline network, it is necessary to open or shut the pipeline valve to block the link or to provide detour route. Also, when there is a necessity to change the quantity of oil and gas in pipeline, the valves should be damped through remote control.

5.5.2 Source

Qualcomm

KT

5.5.3 Actors

Oil and gas pipeline meters, valve controllers, cellular networks, backend servers, remote monitoring, management and control software

5.5.4 Pre-conditions

Cellular network connectivity, Satellite connectivity

5.5.5 Triggers

New pipeline sensor data requiring transport to a backend server

Network dynamic access constraint or network utilization constraints or prior network access policy constraints or device energy minimization considerations can cause delay tolerant sensor data to be buffered (and aggregated if needed) at the gateway and transmitted at a later time

Processing of recent measurements can result in remote requests for additional or more frequent measurements

A firmware upgrade becomes available that needs to get pushed to the gateways

5.5.6 Normal Flow

Sensor data related to oil/gas quantity and quality, pressure, load, temperature, and consumption data is forwarded to backend server that is processed by a remote monitoring service associated with the oil and gas pipeline. Pipeline sensors and pipeline cellular gateways can communicate with each other wirelessly (if sensors and gateways are different nodes in the system). Pipeline cellular or satellite gateways can serve as aggregation points. Sensor data may be locally forwarded until it reaches a gateway or directly transmitted to the gateway depending on proximity of the sensor(s) to each gateway on the pipeline.

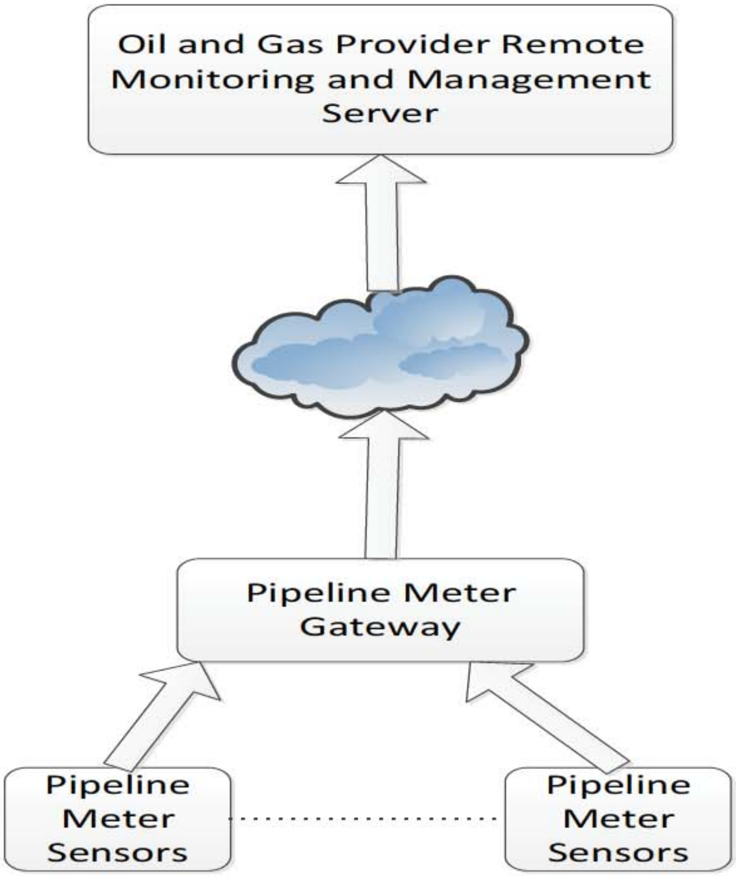


Figure 5-9 Flow - Oil and Gas Pipeline Gateway

5.5.7 Alternative flow

Alternate Flow 1

Pipeline meter data can be stored, aggregated, and forwarded at an appropriate time based on network availability constraints or policy constraints or energy minimization constraints for the pipeline meter gateway. Transmission policies can be designed made to minimize network overhead.

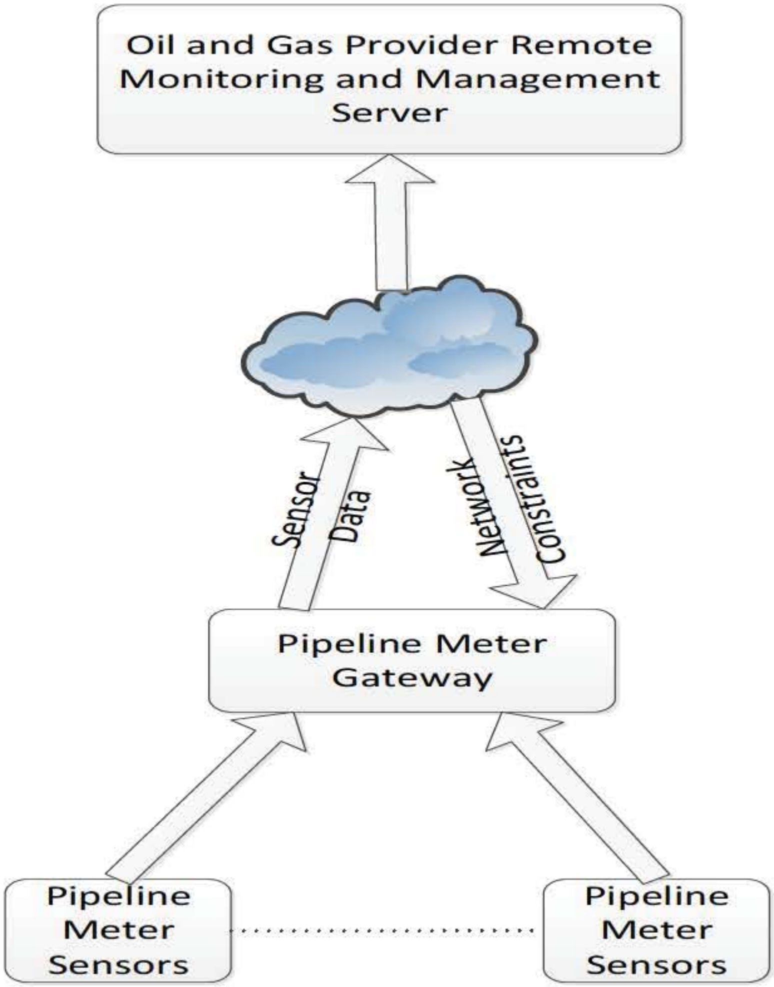


Figure 5-10 Alternate Flow 1 - Oil and Gas Pipeline gateway

Alternate Flow 2

Pipeline meter data can be processed by the remote monitoring and management service. If any anomalies are detected, additional measurements could be triggered, or more frequent measurements could be triggered, or measurements by additional sensors can be triggered by the remote service manager. Firmware upgrades can also be provided by the remote management service. Remote measurement requests are typically triggered or polled only as absolutely needed so as to avoid the overhead of unnecessary polling and network congestion using such schemes with Normal Flow or Alternative Flow 1 preferred for reporting sensor data.

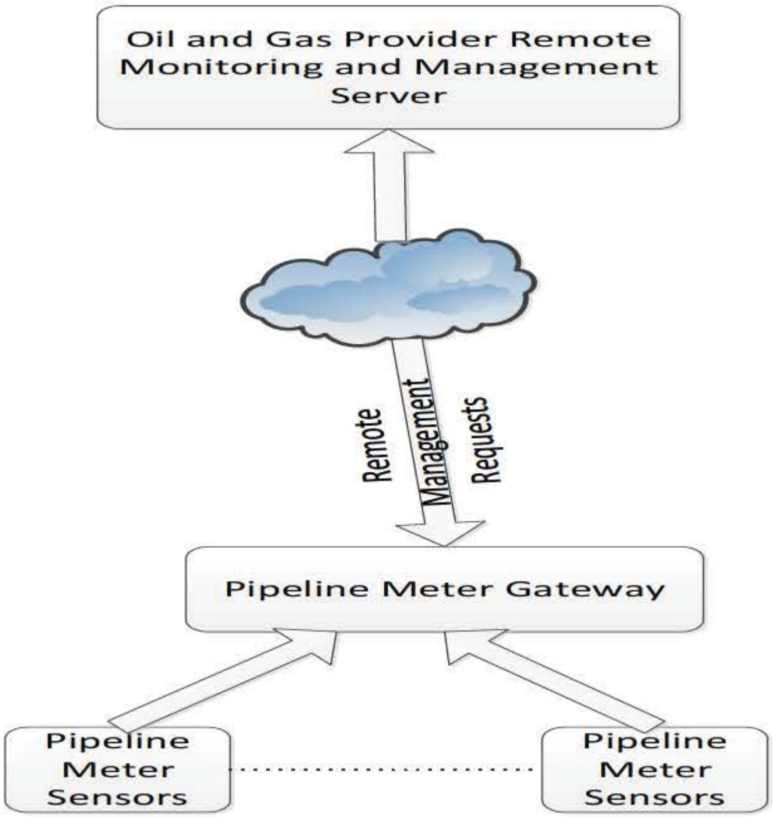

Oil and Gas Provider Remote Monitoring and Management Server
Remote Management Requests
Pipeline Meter Gateway
Pipeline Meter Sensors
Pipeline Meter Sensors

Figure 5-11 Alternate Flow 2 - Oil and Gas Pipeline gateway

Alternate Flow 3

Valve control data should be delivered in real-time. For this purpose, Pipeline Meter Gateway can be used to transport valve control data as well. The Gateway should be connected to and control the targeted valve controllers.

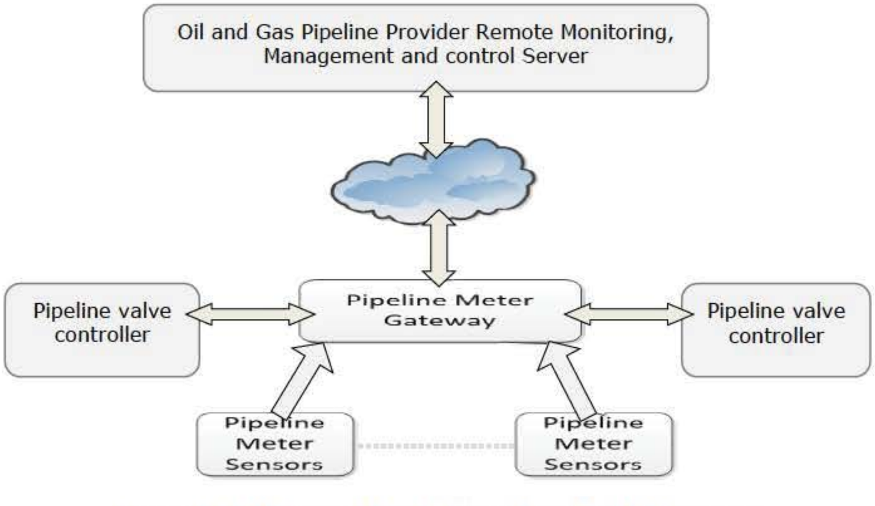

Oil and Gas Pipeline Provider Remote Monitoring, Management and control Server
Pipeline valve controller
Pipeline Meter Gateway
Pipeline valve controller
Pipeline Meter Sensors
Pipeline Meter Sensors

Figure 5-12 Alternate Flow 3 - Oil and Gas Pipeline gateway

5.5.8 Post-conditions

Sensor data is stored in a database associated with the backend server. Remote monitoring service verifies the status of the different pipeline meters.

1. Alternative flow 1

   Data is buffered and transmitted when the network or policy constraints or energy optimization constraints allow transmission of delay-tolerant pipeline sensor data

2. Alternative flow 2

   More frequent or additional measurement request events can get triggered from the network based on processing of recent measurement data.

3. Alternative flow 3

   When a valve controller received errored information from the gateway, the valve controller should send a request of retransmission to the gateway.

5.5.9 High Level Illustration

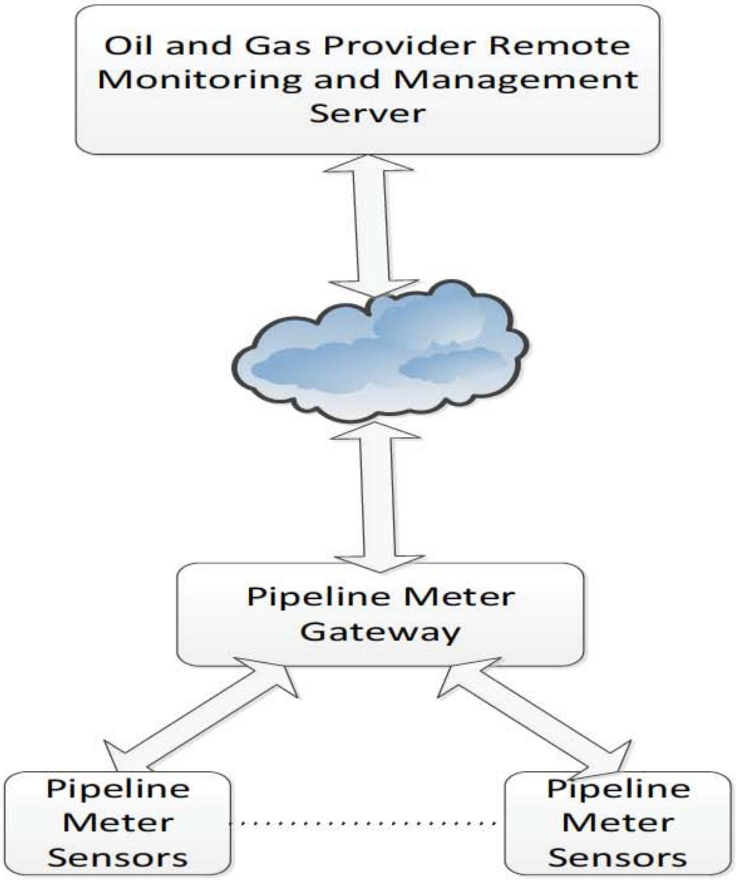


Figure 5-13 High Level Illustration - Oil and Gas Pipeline Gateway

5.5.10 Potential Requirements

Rationale

This use case sets out from the presence of a gateway between one or more oil and gas pipeline sensor(s) and a backend server. One gateway node may serve multiple pipeline sensors and data may be forwarded multihop until it reaches a gateway. Data mules can collect data and dump the information at a gateway for transportation. The ability to locally forward data wirelessly between nodes to a local aggregation point serving as a gateway may be desirable depending on the location of sensor nodes and gateway nodes. Even though the use case is assuming a cellular/satellite gateway, this restriction is not needed in general.

Resulting requirement:

1. The M2M system shall be capable of supporting gateway nodes that are capable of transporting sensor measurements to back end servers.

2. The M2M system shall be capable of supporting static or mobile peer forwarding nodes that are capable of transporting sensor measurements to a gateway node.

Rationale

Pipeline sensors can measure data at predetermined times. Pipeline sensors can also take measurements at random times or based on a request from a backend server to study the health of the pipeline. Therefore, new measurement data may become available at any time. When measurement data is available, the data can be processed locally to understand the criticality of the information. Based on the criticality/urgency of the information, the data can be transported over the network immediately or in a delay-tolerant manner. If an anomaly is detected with regard to the measured data, more frequent measurements may be taken locally or requested from the backend server, to continually assess the criticality of the situation. In case there is no new or relevant information, the system may choose not to transport unnecessary data to reduce network or reduce device energy usage.

Resulting requirement:

3. Whenever a pipeline sensor has measurement data available, it shall be possible for the sensor to send a request to the local pipeline gateway to transport new measurement data to the backend server.

4. Whenever measurement data is available, it shall be possible for the pipeline sensor or a local processing node/gateway to process the information and assess the urgency or criticality of the information, and tag the data appropriately to be critical/urgent or delay-tolerant.

5. Whenever measurement data is available that is determined to be critical/urgent, it shall be possible for the local gateway to send the information to a backend server as soon as possible (such as within in a few 100s of ms). Delay-tolerant data shall be transported within the delay tolerance specified.

6. Whenever measurement data is available that is determined to be not important, the system may choose to not transport the data to reduce network usage or to reduce device energy usage.

7. More frequent measurements may be taken such as when one or more anomalies are detected in the system, which can result it more data and more frequent urgent transmissions in the system, depending on the criticality of the data.

Rationale

Local analytics service functions can be executed to process sensor information. A service function could consist of evaluation rules based on sensor data, and decisions based on rules associated with the data. An evaluation engine can process the rules to then decide whether/when to transmit data. Analytics processing can also be done in a distributed manner, with additional processing on the backend server, or configurability of the evaluation rules at the local gateway by the backend server.

Resulting requirement:

8. A local analytics service function can be executed on the local processing gateway based on evaluation rules associated with the measurement data, and decisions can be taken based on the processing.

9. A distributed analytics service function can be executed in collaboration with a backend server, where additional processing of data can be performed at the backend server, or where the rules associated with local processing can be configurable by a backend server.

Rationale

Incoming requests from the pipeline sensor to the pipeline gateway may not result in immediate forwarding of the data to the backend server if any of the following is applicable: Dynamically changing cellular network availability (coverage); cellular network utilization constraints (policies); device energy consumption or memory constraints. In one of the flows also the quality of the data to be transported (alert=high priority) was relevant for determining when the connection needs to be triggered. Categorization of traffic such as abnormal/urgent data such as a pipeline failure, versus normal traffic can be done at the gateway. Tagging and processing such traffic differently based on application/network/device constraints can be done at the local processing gateway. The system should allow a provisioning policy for handling categorized traffic at the local processing gateway. In many cases, in oil and gas pipeline systems, it is desirable to avoid unnecessary polling of the sensors and minimized network usage. Therefore it is desirable to enable to the system to determine policies for transmitting data such as a scheduled transmission versus an aggressive polling request based on the urgency of information, or aggregating information based on delay tolerance, to best utilize network resources.

Resulting requirements:

10. The local pipeline gateway needs to be capable to buffer incoming requests from the pipeline sensor for transporting data to the backend server and support forwarding them at a later time – which could potentially be a very long time in the order of hours, days or even more – depending on cellular network availability, cellular network utilization policies, device constraints

11. The local pipeline gateway needs to be capable to accept parameters with incoming requests from the pipeline sensor which define a delay tolerance for initiating the delivery of the sensor measurements or parameters for categorizing sensor measurements into different levels of priority/QoS.

12. The local pipeline gateway needs to be cable of receiving policies which express cellular network utilization constraints and which shall govern the decision making in the gateway when initiating connectivity over cellular networks.

13. The local pipeline gateway needs to be capable to trigger connections to the cellular network in line with the parameters given by the request to transport data and in line with configured policies regarding utilization of the cellular network.

14. The local pipeline gateway shall have the ability to categorize the data based on the abnormality/urgency or delay tolerance of the data.

15. The local pipeline gateway can be provisioned with policies to handle categorized traffic.

Rationale

The use case also describes a flow in which the backend server could initiate an action on the local pipeline gateway. The action could include a request for a measurement, or a firmware upgrade push to the gateway, or a change in the policies associated with data transportation. In particular, the ability to provide remote firmware upgrades or remote provisioning of policies is particularly desirable for these pipeline gateways at remote locations.

Resulting requirements:

16. The M2M system shall support transport of data from the backend server to the local pipeline gateway.

17. The M2M system shall support of triggering a cellular connection to the local pipeline gateway in case the gateway supports such functionality

6 Enterprise Use Cases

6.1 Smart Building

6.1.1 Description

Smart building is a M2M service that utilizes a collection of sensors, controllers, alerter, gateways deployed at the correct places in the building combined with applications and server resides on the Internet to enable the automatic management of the building with just limited human labour. Smart building system can greatly reduce the cost involved in managing the building like energy consumption, labour cost. With the smart building system, services like video monitor, light control, air-condition control and power supply can all be managed at the control centre. Some services can be triggered automatically to save the precious time in case of fire, intruder, gas leak etc.

6.1.2 Source

Huawei Technologies UK (ETSI)

Huawei Technologies Co.,Ltd (CCSA)

NEC Europe Ltd. (ETSI)

6.1.3 Actors

M2M Service Provider: A company that provides M2M service including entities like gateway, platform and enables the communication between them. The M2M Service Provider also exposes APIs for the development of all kinds of applications. The gateway provided by the Service Provider can be used to connect to different devices such as sensors, controllers.

Control Centre: The manage centre of the building, all data collected by the sensor is reported to the Control Centre and all commands are sent from the Control Centre. The Control Centre is in charge of the controlling of the equipments deployed around the building.

Smart Building Service Provider: A company that provides smart building services. A Smart Building Service Provider is a professional in the area. It is in charge of install the device all around the building, set up the Control Centre and provide the application that is used to manage the Control Centre and necessary training to workers in the Control Centre on how to manage the system. The Smart Building Service Provider has a business contract with the M2M Service Provider in utilizing the communication, gateway, M2M platform and APIs provided by the M2M Service Provider.

6.1.4 Pre-conditions

The Smart Building Service Provider establishes a business relationship with the M2M Service Provider in using the gateway, M2M platform and APIs.

The Smart Building Service Provider installs all the sensors, controllers, alerter in and around the building and sets up the Control Centre in the building with the application to run the system.

The Control Centre belongs to an estate management company and takes charge of several buildings all over the city. The building in the use case is one of them.

6.1.5 Triggers

None

6.1.6 Normal Flow

1. The light control of the building

The Control Centre needs to control the light in the building by different areas and different floors. The Control Centre also needs to switch on and off all the light in the building. For the management of the lights, the Smart Building Service Provider deployed one gateway in each floor to get connection with the lights in the same floor. Each floor of the building has at least 100 lights and the building has 50 floors above the ground and 5 floors under the ground and each light can be switched separately. The lights in every floor is connected with the gateway using local WIFI network, the gateway is connected with the M2M platform using paid 3GPP network, the Control Centre is connect with the M2M platform using fixed network. A patrolling worker with a mobile device can access to the gateway's local network to switch the lights. The illustration can be seen in figure 6.1

In order to switch the light from the whole floor, instead of sending request from the Control Centre 100 times, the Control Centre creates a group on the gateway of each floor to include all the light on that floor. As a result, the Control Centre could switch the light of a whole floor just by sending one request to the group created on the gateway, the gateway fans out the request to each light to switch them off.

In order to switch the light of the building, instead of sending request from the Control Centre 5500 times, the Control Centre could create a group on the M2M platform to include all the groups created on each gateway on each floor. In this way, the Control Centre simply send one request to the group on the M2M platform, the group fans out the request to the group on every gateway, the group on the gateway fans out the request to each lights to switch it.

The maintenance of the member of the group is the duty of a worker with a mobile device. Whenever a new light is installed, the worker adds the light to the group of the corresponding floor. Whenever a broken light is removed, the worker with the mobile device first searches the light from the group and removes the light from the group.

The Control Centre creates the group in the purpose of controlling the lights, so the group is configured to accept lights only in case the group may cause unexpected result on other devices introduced to the group by mistake. For example, if the type of the group is configured as “light”, then “wash machine” cannot be a member of the group. Because the commands to wash machine is much more complicated. If a wash machine is added to the group of lights by mistake, it may cause unexpected behavior to the wash machine.

The add and remove of the members of the group of each floor is not necessary to be known to the Control Centre, but the Control Centre do know how to switch off the lights from the whole floor. In this way the Control Centre is exempt from the trivial task of maintaining each single light. However in the mean time, the administrator of the Control Centre can always make a list of all the lights and view their status from the Control Centre by retrieving from the group.

2. Intruder

   With the deployment of smart building system, the number of patrollers is greatly reduced. For the security reason, a number of motion detector and cameras are installed all over the building.

   The motion detector and the cameras are configured to work together. During the period when certain floor of the building is in safe mode, whenever the motion detector detects a moving object, the camera captures a picture of the moving object immediately. The picture is sent to the Control Centre for the inspector to verify if it is an intruder or an automated image recognition system. As a result of fast reaction, the motion detector must trigger the photo shot as soon as possible.

   If the inspector sitting in the Control Centre finds that the object captured in the photo is a dog or a cat, he could just ignore the picture. If the figure caught in the picture is a stranger with some professional tools to break into a room. The inspector could send out a security team as soon as possible to the location based on the location reported from the motion detector.

3. Fire alarm

In case of an emergency, the residents of the building need to be evacuated immediately. All the devices related to a fire alarm need to be triggered almost at the same time. Whenever the fire sensor detects a fire in the building, a chain group of devices associated with the fire detection shall be turned on simultaneously such as the siren, the evacuation guide light, start the water pouring system, stop the elevator, cut off the electricity at certain areas, send message to the hospital, call the fireman, in a way not interrupting each other. Due to the possible latency and unavailability on the network to the Control Centre, the trigger of the devices on one floor is configured in the gateway.

If only one fire sensor in one room of the building detects a fire with a range less than one square meter, siren and water pouring system in the room would be switched on to alarm the resident to put out the fire. If lots of fire sensors all detect fire together with smoke sensors, temperature sensors reporting unusual situations, the whole fire alarm system will be triggered and all the residents in the building will be evacuated. If in the mean time of a fire alarm, the sensors detect that the temperature is below the threshold which means the fire is under control, the alarm can be cancelled automatically to all sirens and actuators to avoid the panic.

With the configuration on the gateway, the trigger of the devices can be very fast so that the damage caused by the fire can be limited to its minimum

6.1.7 Alternative flow

None

6.1.8 Post-conditions

None

6.1.9 High Level Illustration

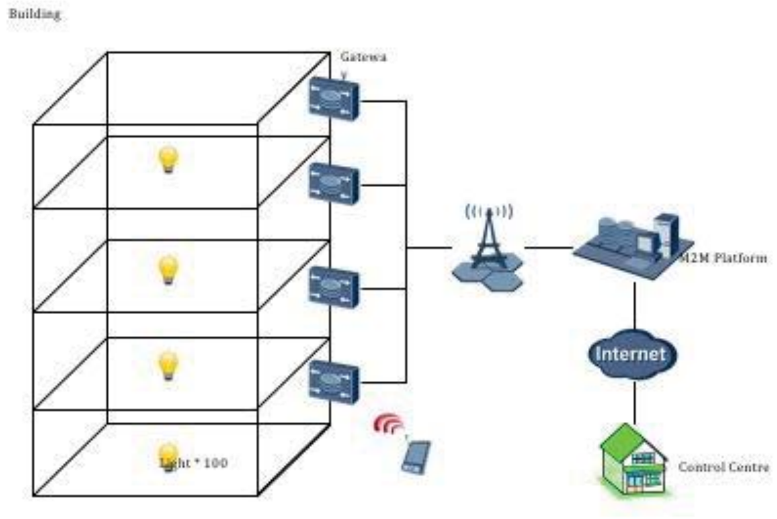


Figure 6-1 Smart Building Scenario

6.1.10 Potential Requirements

1. The M2M system shall support the action chain harmonize a series of actions among a group of between devices, in a way not interrupting each other.
2. The M2M system shall harmonize a series of actions based on certain conditions that support the action chain between devices shall subject to certain conditions.
3. The M2M system shall support the devices to report their locations.
4. The M2M system shall support a mechanism to group a collection of devices together.
5. The M2M system shall support that same operations can be dispatched to each device via group.
6. The M2M system shall support the members' management in a group i.e. add, remove, retrieve and update.
7. The M2M system shall support that the group can check if its member devices are of one type.
8. The M2M system shall support the group to include another group as a member.

7 Healthcare Use Cases

7.1 M2M Healthcare Gateway

7.1.1 Description

This use case addresses a healthcare gateway to transport healthcare sensor data from a patient to a backend server, and to also support bidirectional communications between a backend server via a gateway. The use case results in a set of potential requirements out of which some are specific to the fact that cellular connectivity is assumed between gateway and backend. Other than that, this use case is not restricted to cellular connectivity.

This use case also addresses the situations where some of M2M System components are not available due to, for example, disaster

7.1.2 Source

Qualcomm.

Several scenarios also supported by guidelines [i.14] defined in Continua Health Alliance should be covered by this use case.

Samsung SDS (as for the alternative flow with some components of the M2M System in failure)

7.1.3 Actors

- Patients using healthcare sensors
- Health-care gateways (also known as AHDs (Application Hosting Devices) in Continua Health Alliance terminology). Examples of healthcare gateways can include wall plugged devices with wired or wireless connectivity, or mobile devices such as smartphones.
- Operating healthcare service enterprise backend servers (equivalent to a WAN Device (Wide Area Network Device) in Continua Health Alliance terminology)
- Health care providers, operating healthcare enterprise backend servers

- Care givers and authorized users that could eventually access health sensor data
- Wide Area Network operator

7.1.4 Pre-conditions

- Operational healthcare sensor(s) that requires occasionally or periodically transport of sensor data to a backend server.
- A local healthcare gateway is available that can be used to transport data from the healthcare sensor to a backend server. It is open as regards who owns and/or operates this local gateway. Different scenarios shall be possible supported (patient, healthcare provider, care-giver, M2M service provider, wide area network operator).
- Network connectivity is available for transporting healthcare sensor data from the local gateway to the backend server.
- A backend server that is hosting applications to collect measurement data and makes it available to care-givers, healthcare-providers or the patient.

7.1.5 Triggers

The following triggers could initiate exchange of information according to the flows described further-below:

- Patient-initiated measurement request (Trigger A). In this case, the patient decides to take a measurement and triggers the processing in the system.
- Static configured policy at a healthcare gateway that requests patient to initiate measurement (Trigger B). This can be an explicit message from the gateway device to a patient device, or it could just a indicator on the gateway itself such as a pop-up message or an indicator light requesting measurement.
- Static configured policy at a healthcare gateway that directly requests sensor data without patient intervention (Trigger C). This can be used in conjunction or in lieu of Triggers A or B. Some sensor data may be measurable or accessible without patient intervention so that the gateway merely needs to communicate with one or more sensors to obtain the data.
- Patient monitoring app on healthcare service backend server that triggers generation of sensor data (Trigger D).
- Dynamic patient monitoring request from the healthcare service provider (Trigger E).
- Availability of new patient healthcare data at a healthcare gateway that requires transport to a backend server.
- Availability of new patient healthcare data at a backend server that requires sharing with authenticated users such as a nurse/doctor (healthcare provider) and a patient’s relative (such as a child care-giver).
- Health care service provider needing to contact patient to take measurements.
- Analysis of healthcare patient sensor info or trends that triggers the need to take action on behalf of patient (for example determination of a deteriorating health condition).
- QoS-aware data buffering policy on the healthcare gateway.
- Network-aware and/or device-aware delay-tolerant data management policy on the healthcare gateway. Network dynamic access constraints or network utilization constraints or prior network access policy constraints or device energy minimization considerations can cause delay tolerant sensor data to be buffered (and aggregated if needed) at the gateway and transmitted at a later time.

- Failure in the components of the M2M System for the healthcare service. (e.g. functional failure in Wide Area Network, functional failure in Healthcare Service Backend Server).

The following clauses describe different flows that are possible in the m2m healthcare gateway system. For each flow, the events corresponding to the flow are high-lighted in the corresponding figure. Other events may be shown in a figure that are preserved to reflect the different types of processing that can occur in the system, with new events added in each subsequent figure to increase the complexity of the system. The high-level illustration in 7.1 provides a comprehensive summary description of the overall system.

7.1.6 Normal Flow

A measurement of the healthcare sensor is initiated as shown in 7-1. Patient can initiate the generation of sensor data such as taking a glucose meter measurement (Trigger A). The measurement may also be initiated based on some pre-defined schedule.

1. at the healthcare gateway (Trigger B or C).

2. The healthcare sensor data is forwarded to a backend server by a healthcare-gateway. If the data has a QoS indicator such as dynamic latency/bandwidth and/or delay tolerance, the gateway can determine whether to send the data immediately, or whether to buffer and send the data at a later time. Buffered data can be aggregated with past data or future data for a future aggregated transmission over the network. In wireless/cellular networks, aggregated transmissions can reduce the utilization of the network by requesting access to the network less frequently.

3. Measured data (or processed/interpreted versions of the data) that arrives at the healthcare service enterprise backend server may need to be forwarded to authorized subscribers – such as family care-giver or a nurse/doctor – via notifications. Subscriptions can be set up in advance, and configured at the backend server, so that when the data arrives, the subscribers can be notified. Filters can be associated with the subscriptions, so that only selective data or alert information can be sent to subscribers.

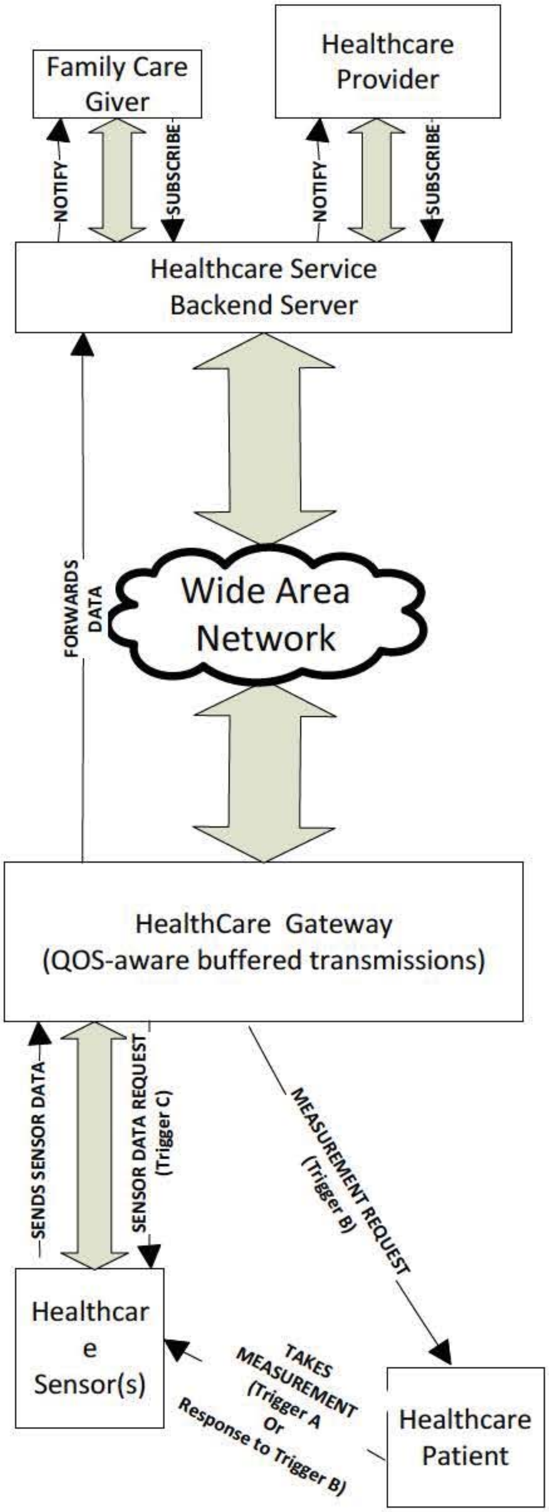

Family Care Giver
Healthcare Provider
NOTIFY
SUBSCRIBE
NOTIFY
SUBSCRIBE
Healthcare Service Backend Server
FORWARDS DATA
Wide Area Network
HealthCare Gateway (QOS-aware buffered transmissions)
SENDS SENSOR DATA
SENSOR DATA REQUEST (Trigger C)
MEASUREMENT REQUEST (Trigger B)
Healthcare Sensor(s)
TAKES MEASUREMENT (Trigger A Or Response to Trigger B)
Healthcare Patient

Figure 7-1 Healthcare Measurement Data Processing Flow

7.1.7 Alternative flow

Alternative Flow 1– Network/Device-aware transmissions

The flow in figure 7-2 depicts network/device-aware constraint processing in the system. This flow is the same as the regular flow with the following exceptions: The healthcare sensor data may need be stored on the gateway and forwarded at a future time based on one or more of the following factors:

- delay tolerances associated with the data.
- network policy constraints (efficiency, avoidance of peak loads, protection of spectrum).
- device constraints (energy consumption, data tariff).
- temporary lack of coverage of network connectivity.

Multiple measurements can be aggregated and transmitted together at a future time.

Measurements can be taken with or without patient intervention and sent to the healthcare gateway. As measured data arrives at the healthcare gateway, its QoS indicators such as dynamic latency/bandwidth and delay tolerance can be processed. Delay tolerant data can be buffered and aggregated with past and future delay-tolerant data, with network/device-aware constraints can applied to determine an appropriate time to transmit the data.

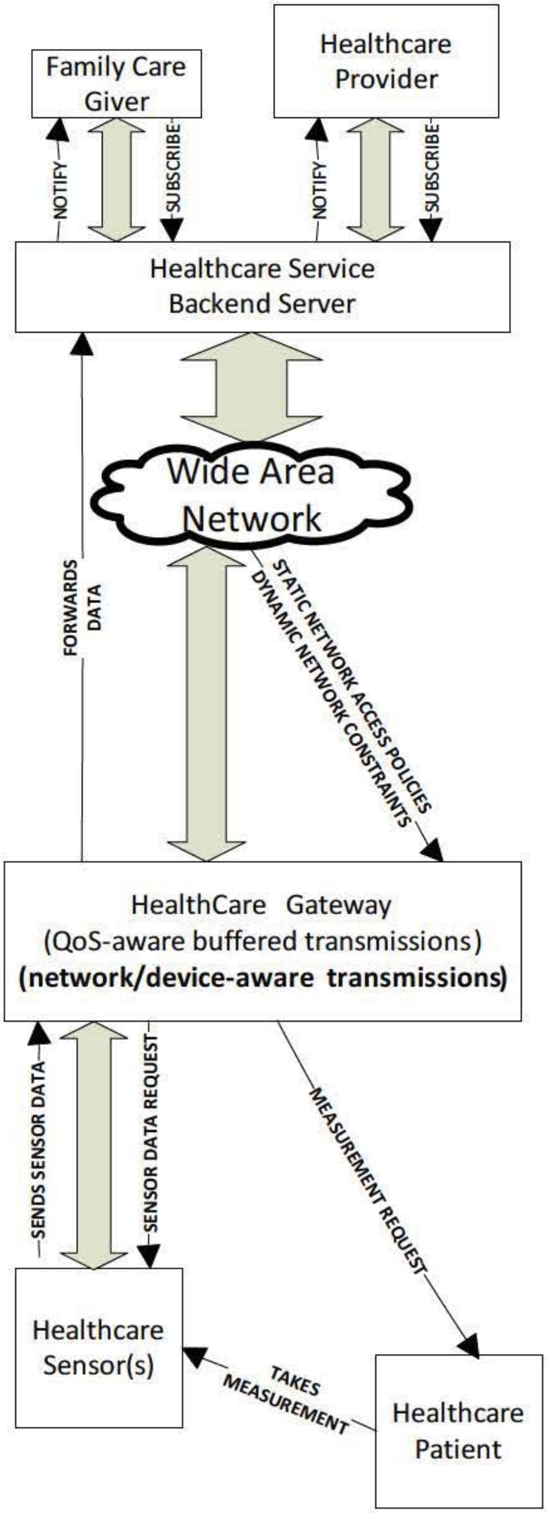


Figure 7-2 Network/Device-aware Flow

Alternative Flow 2– Remote Monitoring

Figure 7-3 depicts the event flow for remote monitoring from the healthcare service enterprise backend server. The backend server may expect the patient to submit sensor data periodically or with a pre-defined schedule. In the absence of a typically expected sensor data event, the backend server can trigger an event to request the patient to take a measurement.

In this case, the trigger (Trigger D) arrives over a wide-area-network from the patient monitoring app on the healthcare service backend server delivered to the healthcare gateway. The patient monitoring app could generate this request based on a statically configured policy to request measurements or due to some dynamic needs based on processing of previous patient data.

Optionally, the healthcare service provider may generate a measurement request (Trigger E) that can be received by the patient monitoring app on the backend server, which can subsequently submit a request over the wide area network for the patient monitoring request to the healthcare gateway.

The healthcare gateway forwards the received request to the patient. In many cases, it is possible that a device associated with the patient, such as the healthcare cellular gateway, or a smartphone connected to the gateway, does not always have an active network connection, and that such a device may be asleep. In such a case, the measurement request can arrive with a wakeup trigger (such as using an SMS) (also called “shoulder tap” in Continua Health Alliance terminology) to the healthcare gateway, which can then establish connectivity with the backend server to determine the purpose for the trigger, and then subsequently process the patient measurement request.

The patient subsequently takes the sensor measurement upon receiving the request. Alternatively, some sensor measurements could be taken without patient intervention. Measured sensor data is then received at the healthcare gateway, and subsequently transmitted based on processing the QoS/Network/Device-aware constraints for transmission.

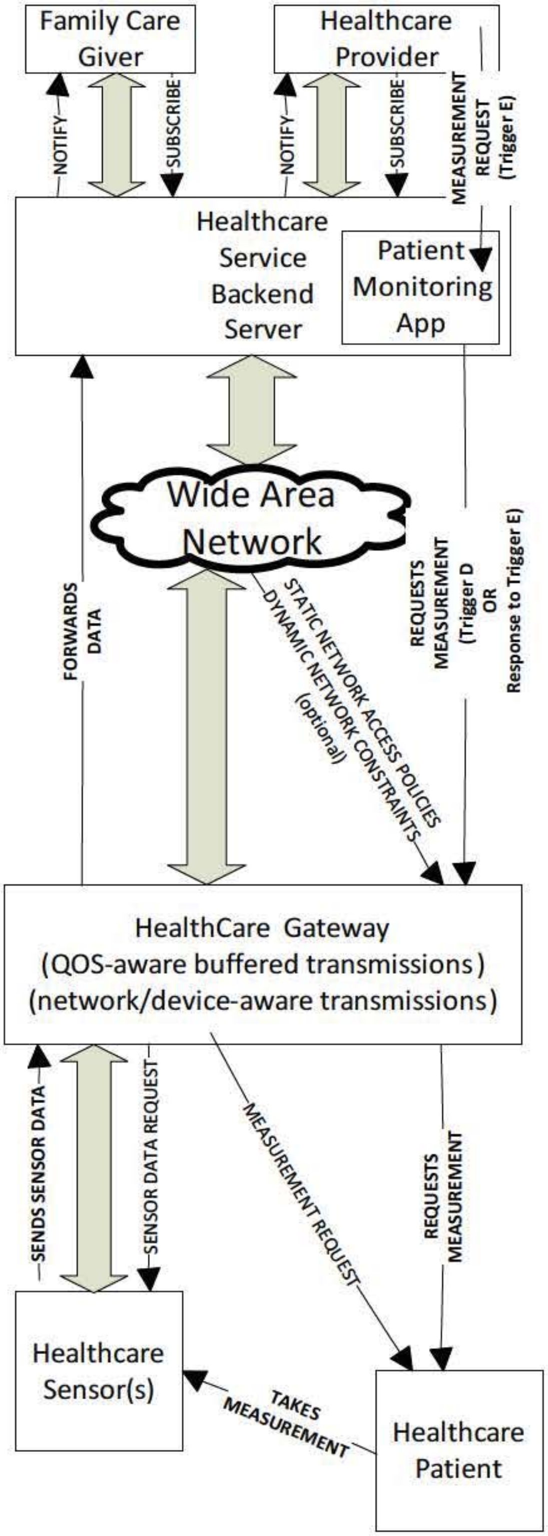


Figure 7-3 Remote Monitoring Flow

Alternative Flow 3 Local Gateway Data Analysis

Figure 7-4 illustrates a Local Gateway Data Analysis flow of events. The local gateway node can continuously process the data that it forwards. It can have smart algorithms to detect health anomalies associated with the patient. In case no anomalies are detected, the health sensor data may only be forwarded occasionally (see also alternative flow 1). In case an anomaly is detected, the local gateway needs to send an alert to the health care provider or the care-giver or to the patient if desired.

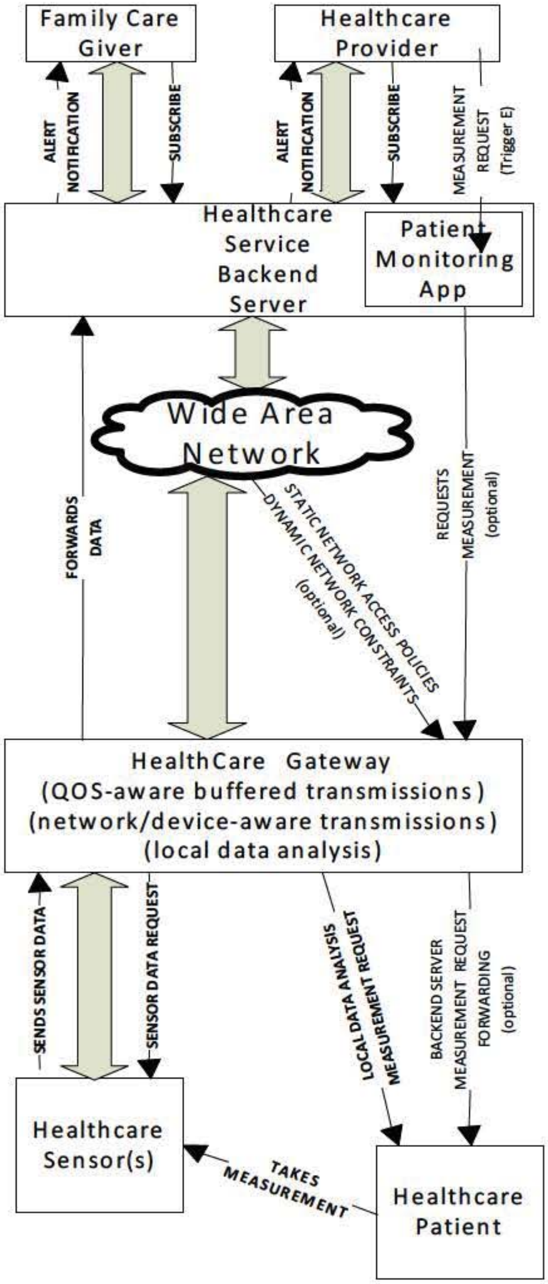

Family Care Giver
Healthcare Provider
ALERT NOTIFICATION
SUBSCRIBE
ALERT NOTIFICATION
SUBSCRIBE
MEASUREMENT REQUEST (Trigger E)
Healthcare Service Backend Server
Patient Monitoring App
Wide Area Network
FORWARDS DATA
STATIC NETWORK ACCESS POLICIES
DYNAMIC NETWORK CONSTRAINTS (optional)
REQUESTS MEASUREMENT (optional)
HealthCare Gateway (QOS-aware buffered transmissions) (network/device-aware transmissions) (local data analysis)
SENDS SENSOR DATA
SENSOR DATA REQUEST
LOCAL DATA ANALYSIS MEASUREMENT REQUEST
BACKEND SERVER MEASUREMENT REQUEST FORWARDING (optional)
Healthcare Sensor(s)
TAKES MEASUREMENT
Healthcare Patient

Figure 7-4 Local Gateway Data Analysis Flow

Alternative Flow 4 – Partial Failure Case

Figure 7-5 illustrates a partial system failure, i.e. the failure of Healthcare Service Backend Server and/or the failure of the connection between Healthcare Gateway and Wide Area Network. In this situation, nevertheless, components of the healthcare system that are not in failure should continue their normal operations. Examples of the 'normal operation' are as follows:

1. Reports from Healthcare sensor are received by and stored in Healthcare Gateway

2. Notification from Healthcare Gateway (e.g. Measurement triggers) is forwarded to Patient

3. If the messages transmitted between Healthcare Sensors and Healthcare Gateway were encrypted before the failure for the privacy of patients, that encryption should be maintained after the failure. (c.f. For maintaining the security mechanism in an isolated domain, a locally operable key management mechanism can be introduced.)

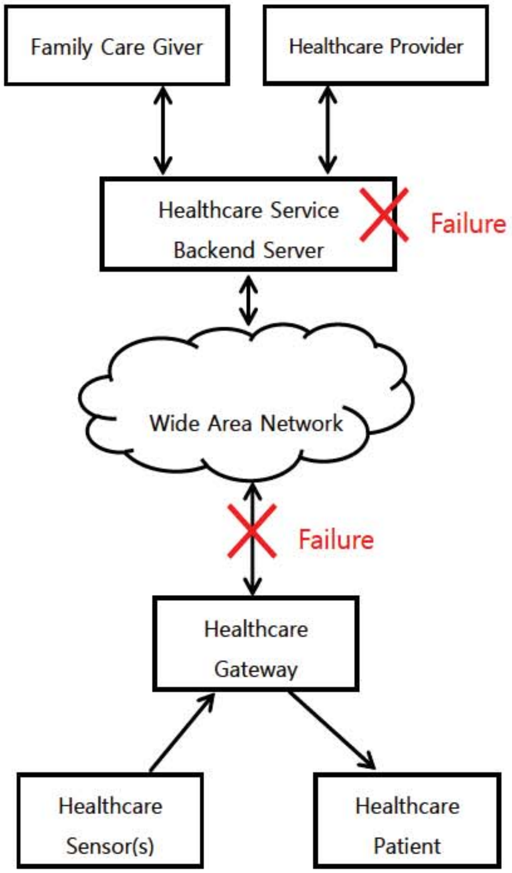


Figure 7-5 Example of failures in components of the M2M System for healthcare service

7.1.8 Post-conditions

1. Normal flow

   Sensor data is stored in a database associated with the backend server. Healthcare provider and care-giver observe data to ascertain status of patient’s health.

2. Alternative flow 1

   Data is buffered and transmitted when the network constraints or policy constraints or device energy minimization constraints allow the transmission of delay-tolerant data.

3. Alternative flow 2

   Patient takes measurement and sends data to backend server.

4. Alternative flow 3

   Local data analysis with indication of abnormal condition results in an alert message sent to the health care provider and optionally to the patient.

5. Alternative flow 4

   Components of the healthcare system that are not in failure continue their normal operations.

7.1.9 High Level Illustration

Figure 7-6 summarizes the overall description of this use-case. All the flows and connectivity should be self-explanatory based on the discussions in the previous clauses.

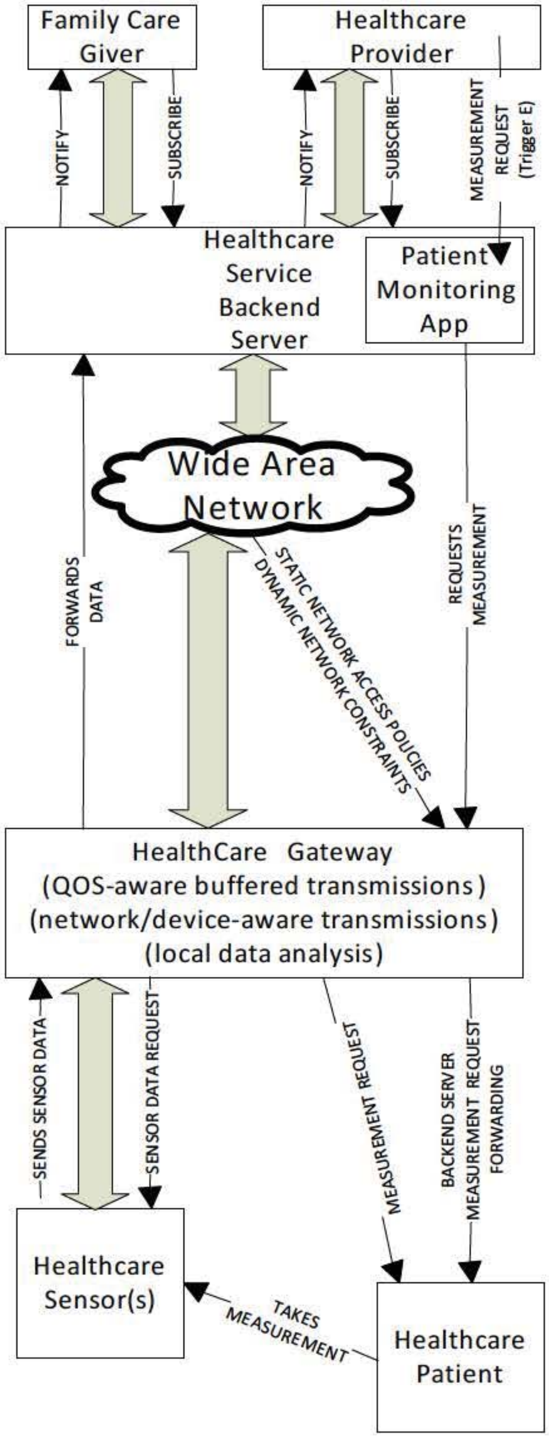


Figure 7-6 Healthcare Gateway High Level Illustration

7.1.10 Potential Requirements

Rationale

This use case sets out from the presence of a gateway between one or more healthcare sensor(s) and a backend server. Even though the use case is assuming a cellular gateway, this restriction is not needed in general. Resulting requirement:

1. The M2M system shall be capable of supporting gateway nodes that are capable of transporting sensor measurements to back end servers.

Rationale

Sensors can measure patient data with or without patient initiation. Therefore, new measurement data may become available at any time. Resulting requirement:

2. Whenever a healthcare sensor has measurement data available, it shall be possible for the sensor to send a request to the local healthcare gateway to transport new measurement data to the backend server.

Rationale

Incoming requests from the healthcare sensor to the healthcare gateway may not result in immediate forwarding of the data to the backend server if any of the following is applicable: Dynamically changing cellular network availability (coverage); cellular network utilization constraints (policies); device energy consumption or memory constraints or mobility, and data delay tolerance/QoS information. In some cases, the delay tolerance may be very low (implying requiring immediate transport) whereas in other cases, the delay tolerance can be significant. In some other variants where real-time delivery or near-real-time delivery is of interest, then real-time latency and bandwidth QoS requirements become significant. More than one healthcare sensor may provide data at the same time, so that the healthcare gateway will need to process one or more concurrent data streams. Event categories associated with the data to be transported (such as alert=high priority) can also be relevant for determining when the connection needs to be triggered.

Resulting requirements:

3. The local healthcare gateway needs to be capable to buffer incoming requests from the healthcare sensor for transporting data to the backend server and support forwarding them at a later time – which could potentially be a very long time in the order of hours, days or even more – depending on cellular network availability, cellular network utilization policies, device constraints

4. The local healthcare gateway needs to be capable of accepting parameters with incoming requests from the healthcare sensor source which define a QoS policy for initiating the delivery of the sensor measurements or parameters for categorizing sensor measurements into different levels of priority/QoS.

5. The local healthcare gateway needs to be able to concurrently process multiple streams of data from different sources with awareness for the stream processing requirements for each of the streams. The local healthcare gateway needs to address the QoS policy of one or more concurrent streams while taking into account network constraints such as available link performance and network cost. The local healthcare gateway needs to adapt to dynamic variations in the available link performance or network communication cost or network availability to deliver one or more data streams concurrently

6. The local healthcare gateway needs to be capable of receiving policies which express cellular network utilization constraints and which shall govern the decision making in the gateway when initiating connectivity over cellular networks.

7. The local healthcare gateway needs to be capable to trigger connections to the cellular network in line with the parameters given by the request to transport data and in line with configured policies regarding utilization of the cellular network

Rationale

A subscription and notification mechanism was described in this use case. Only authenticated and authorized users (e.g. care-giver, relatives, and doctors) shall be able to subscribe to healthcare sensor measurement data and get notifications and access to the measured data. These authenticated and authorized stakeholders are typically using applications that use the M2M system to access the measured data. Resulting requirement:

8. The M2M system shall be capable of supporting a mechanism to allow applications (residing on the local gateway, on the backend server or on the sensor itself) to subscribe to data of interest and get notifications on changes or availability of that data.

9. The M2M system needs to be able to allow access to data that is being transported or buffered only to authenticated and authorized applications

Rationale

The use case also describes a flow in which the backend server could initiate an action on the local healthcare gateway. Resulting requirements:

10. The M2M system shall support transport of data from the backend server to the cellular healthcare gateway.

11. The M2M system shall support of triggering a cellular connection to the local healthcare gateway in case the gateway supports such functionality.

Rationale

Different subscribers may be interested in different information so that each subscriber may want to get notified only for events of interest to that subscriber:

12. Subscriber-specific filters can be set up at the healthcare service enterprise backend server so that each subscriber can be notified only when information/events relevant to the subscriber are available/occur.

Rationale

The M2M healthcare gateway device can be without an active network connection because it is in a sleep mode of operation to save energy and/or because it is trying to save radio/network resources. A patient monitoring app may be desirous of communicating with the gateway device when the gateway device is in this sleep mode of operation

13. The M2M system shall be able to support a wakeup trigger (aka "shoulder-tap") mechanism (such as using SMS or alternate mechanisms) to wake up the gateway. The gateway can subsequently establish a network connection and query the enterprise backend server for additional information, and the enterprise backend server may then respond with adequate information to enable further processing of its request.

14. When some of the components of M2M System are not available (e.g. WAN connection lost), the M2M System shall be able to support the normal operation of components of the M2M System that are available.

15. When some of the components of M2M System are not available (e.g. WAN connection lost), the M2M System shall be able to support the confidentiality and the integrity of data between authorized components of the M2M System that are available.

7.2 Use Case on Wellness Services

7.2.1 Description

This use case introduces several services based on wellness data collected by wellness sensor devices via mobile device such as smartphones and tablets which is regarded as M2M gateway.

Some wellness sensor devices are equipped with M2M area network module and measure individual wellness data. The mobile device connects to the wellness sensor devices by using the M2M area network technology, collecting and sending the wellness data to application server.

It is important to consider that mobile device as M2M gateway has mobility. For instance, there are possibilities for a mobile device to simultaneously connect to many wearable wellness sensor devices, and to connect newly to wellness sensor devices which have never connected previously at the location of outside.

This use case illustrates potential requirements from the use case of wellness services utilizing mobile device.

7.2.2 Source

KDDI (TTC)

7.2.3 Actors

- M2M Device: wellness sensor device is blood pressure sensor, heart rate sensor and weight scale, for example. It can measure wellness data of users, may be multi-vendor, and equipped with several kind of communication protocol.
- M2M Area Network: network which connects between M2M device and M2M gateway.
- M2M Gateway: mobile device (e.g. a smart phone) which can receive wellness data from wellness sensor devices and communicate with application servers.
- Mobile Network: network which has functions to communicate wellness data and control message between M2M gateway and M2M service platform.
- M2M Service Platform: platform where management server is located and which is used by the Application Server to communicate with the M2M Gateway.
- Management Server: server which manages the gateway such as mobile device, and controls its configuration such as installing/uninstalling applications.
- Application Server: server which serves the wellness services such as indicating the graph of wellness data trend.

Note: Definition of some words is in discussion. Therefore, the description of these actors may change.

7.2.4 Pre-conditions

- Wellness sensor devices are able to establish a connection to the mobile device in order to send wellness data to M2M Service Platform or Application Server.
- It is first time to associate the mobile device with the wellness sensor devices.

7.2.5 Triggers

New wellness sensor devices such as weight scale are detected by mobile device. User tries to associate the detected devices. Examples are below:

- User buys several kind of wearable wellness sensor devices such as blood pressure sensor, heart rate sensor. In order to start monitoring vital data using these sensors, User tries setting of these devices simultaneously. Note that please refer to [i.4] ETSI TR 102 732 “Use cases of M2M applications for eHealth”. (Normal Flow)
- User buys wellness sensor devices such as weight scale, and newly deploys them at User’s house to check the wellness status daily. (Normal Flow)
- User goes to a fitness center to do exercise and checks the effect by utilizing equipment which is owned by fitness center and has never connected to User’s mobile device. (Alternative Flow 1)

7.2.6 Normal Flow

Usually wellness sensor devices are bought by Users. These devices are deployed in User’s house, or are worn with User.

1. The mobile device detects new wellness sensor devices and tries to connect to it under User’s permission to connect (pairing between sensor device and mobile device).
2. The mobile device has established a connection to the wellness sensor device, and then the mobile device receives additional information of the wellness sensor device (e.g. type of device, service certificates of the device, required application software …).
3. The mobile device is provided with the appropriate application software from the Management Server and is appropriately configured by the Management Server.
4. When the User measures the data by using wellness sensor device, the mobile device collects the data and sends it to the Application Server.

7.2.7 Alternative flow

Alternative Flow 1

1. As indicated in the Normal Flow, usually the wellness service collects the data from wellness sensor devices which the User owns.
2. When the mobile device is brought outside, there is an opportunity to connect new wellness sensor devices (e.g. blood pressure which is set in fitness center).
3. The mobile device detects new wellness sensor devices and tries to connect to them under User’s permission to connect.
4. The mobile device has established a connection to the wellness sensor device and then the mobile device receives additional information of the wellness sensor device (e.g. type of device, service certificates of the device, required application software …).
5. The mobile device is provided with the appropriate application software and is appropriately configured by the Management Server.
6. When the User measures the data by using wellness sensor device, the mobile device collects the data and sends it to the Application Server.

Alternative Flow 2

1. The wellness service may be an optional subscriber service to be charged. The User subscribes it and creates an account on the Application Server.
2. When the User utilizes the wellness service, at first the User needs to activate the service on the Application Server.

3. When the mobile device detects wellness sensor devices, it requests the Management Server to provide appropriate application software with configuration to the mobile device.

4. The Management Server checks with the Application Server if the User has subscribed to the service and activated it or not.

5. And then, if the User is not subscribed to the service or has not activated it, the Management Server does not provide any application software.

Alternative Flow 1

After the User has collected the data, the User is able to disconnect the mobile device from the wellness sensor device and to de-activate the service.

1. If the User brings the mobile device out of the range of M2M Area Network, the mobile device disconnects the wellness sensor device automatically.

2. The User is also able to disconnect these devices by operating settings of the mobile device or by waiting for a while until the wellness sensor device disconnect by itself.

3. The User is also able to cancel the optional service. The User applies the cancellation to the Application Server. After the Application Server accepts the cancellation, the Management Server checks with the Application Server. The Management Server confirms the cancellation, it makes application software de-activate and/or remove from the mobile device.

7.2.8 Post-conditions

- Measured wellness data are stored in the M2M Service Platform or the Application Server.
- User is able to access to the Application Server and explore the graph of the wellness data trend.

7.2.9 High Level Illustration

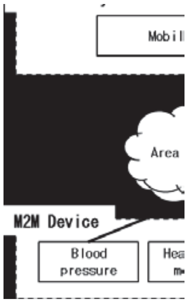


Figure 7-7 Wellness Service High Level Illustration

7.2.10 Potential Requirements

- M2M Gateway SHALL be able to detect device that can be newly installed (paired with the M2M Gateway).
- Upon detection of a new device the M2M Gateway SHALL be able to be provisioned by the M2M Service Platform with an appropriate configuration which is required to handle the detected device.
- The M2M Service Platform SHALL be able to provide an authenticated and authorized application in the M2M Gateway with appropriate configuration data.

7.3 Secure remote patient care and monitoring

7.3.1 Description

E-health applications, that provide the capability for remote monitoring and care, eliminate the need for frequent office or home visits by care givers, provide great cost-saving and convenience as well as improvements. “Chronic disease management” and “aging independently” are among the most prominent use cases of remote patient monitoring applications. More details of the actors and their relationships for these use cases are mentioned in details in an ETSI document [i.4] and are not covered here. Instead this contribution provides an analysis of specific security issues pertaining to handling of electronic health records (EHR) to provide a set of requirements in the context of oneM2M requirement definition work.

Remote patient monitoring applications allow measurements from various medical and non-medical devices in the patient’s environment to be read and analyzed remotely. Alarming results can automatically trigger notifications for emergency responders, when life-threatening conditions arise. On the other hand, trigger notifications can be created for care givers or family members when less severe anomalies are detected. Dosage changes can also be administered based on remote commands, when needed.

In many cases, the know-how about the details of the underlying communications network and data management may be outsourced by the medical community to e-health application/ solution provider. The e-health solution provider may in turn refer to M2M service providers to provide services such as connectivity, device management. The M2M service provider may intend to deploy a service platform that serves a variety of M2M applications (other than e-health solution provider). To that end, the M2M service provider may seek to deploy optimizations on network utilization, device battery or user convenience features such as ability of using web services to reach application data from a generic web browser. The M2M service provider may try to provide uniform application programming interfaces (APIs) for all those solution providers to reach its service platform in a common way. From the standpoint of the M2M application, the application data layer rides on top a service layer provided by this service platform. By providing the service platform and its APIs, the M2M SP facilitates development and integration of applications with the data management and communication facilities that are common for all applications.

As part of providing connectivity services, the M2M service provider may also provide secure sessions for transfer of data for the solution providers that it serves. In many jurisdictions around the world, privacy of patient healthcare data is tightly regulated and breaches are penalized with hefty fines. This means the e-health application provider may not be able to directly rely on the security provided by the M2M service provider links/sessions and instead implement end to end security at application layer. This puts additional challenges on the M2M service platform, since it needs to provide its optimizations on encrypted data.

One particular issue with e-health is that not only the data is encrypted, but it may also contain data at different sensitivity levels, not all of which appropriate to each user. For instance in the US the Health Insurance Portability and Accountability Act (HIPAA) regulates the use and disclosure of protected health information. Different actors within a healthcare scenario may have different levels of authorizations for accessing the data within the health records, so the information system must take care to present the health data to each user according to the level of authorization for that user. A process, common to address this issue is redaction. This means that one starts with a document that originally includes data of all sensitivity levels and then removes any piece of information that has a higher sensitivity level than the pre-determined redaction level (RL). The end result is a redacted version of the initial document that can be presented to a person/entity that has the matching authorization level (AL). Persons with lower AL are not authorized to view this particular version of document. The redaction engine can produce multiple versions of the initial records, where each version corresponds to one redaction level (RL) including material at specific sensitivity level (and lower).

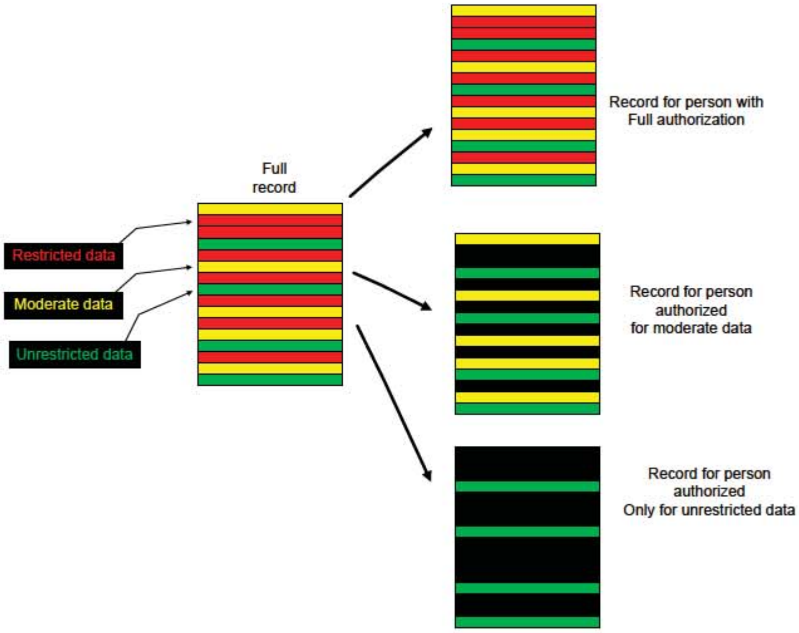


Figure 7-8 – An illustration of a process with 2 levels of redaction. Black color indicates a data field that is masked from an unauthorized user.

Care must be taken to ensure that only authorized users have access to data. Therefore, the system must match the redaction level (RL) of data with the authorization level (AL) and present the proper version of the record for each actor.

The redaction engine may reside at a policy control server or at the application server operated by the M2M application service provider. The policy server may also hold policies on which users get which authorization level (AL), while an authorization server may be in charge of authenticating each user and assigning her the proper AL.

In a system relying on notifications based on prior subscriptions, data must be examined first to determine which subscribers should receive notifications and then only those subscribers should be capable to retrieve the data about which the notification is sent.

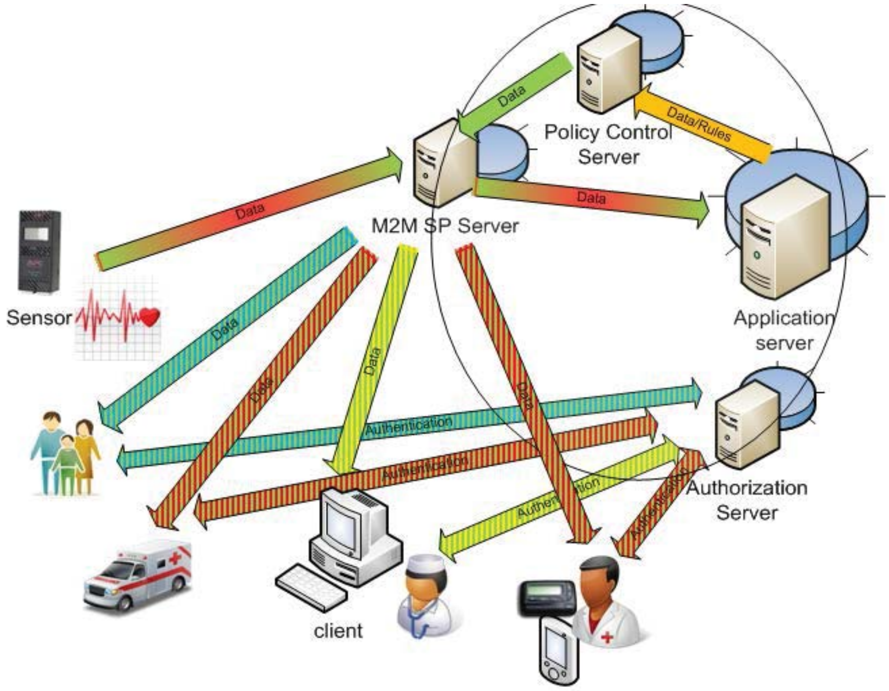


Figure 7-9 An e-Health application service capable of monitoring remote sensor devices and producing notifications and data to health care personnel based on their authorization level.

7.3.2 Source

Motorola Mobility, ETSI member

7.3.3 Actors

- Patients using sensor (medical status measurement) devices
- E-Health application service providers, providing sensor devices and operating remote patient monitoring, care and notification services
- Care givers (e.g. nurses, doctors, homecare assistants, emergency responders) and other administrative users with authorization to access healthcare data (e.g. insurance providers, billing personnel). We also refer to these entities as “participants in the healthcare episode” in some occasions.
- M2M service providers, network operators, providing connectivity services for the patients, e-health application providers and care givers.

7.3.4 Pre-conditions

- A categorization rule set, that is able to categorize various entries within a medical record according to the sensitivity levels and label them accordingly, must exist.
- A redaction engine that is able to examine the raw medical record and produce different versions of the record at different redaction levels (RL) with only data that is at or below a sensitivity level.
- A policy engine that is able to examine medical records and determine level of criticality (applicable to one of the flows described).
- A set of authorization policies that describe what authorization level (AL) is required to be able to access data at each redaction level (RL).
- An authorization engine/server that interacts with each user of the e-health application to verify their claimed AL, for example the server may perform an authentication function with the user.
- The e-health application server that is capable of interacting with the authorization server to check the AL of each user to determine the user’s RL before serving data at the requested (or appropriate) RL to that user.

7.3.5 Triggers

- Creation of new measurement data by a remote medical device.
- Analysis of received measurement data at application servers, and determination of need for redaction, or creation of alarms and notifications, etc.
- Requests from participants in a health care episode (caregivers) for sensitive medical records.
- Arrival of new participants (new doctors, etc) in the health care episode

7.3.6 Normal Flow

In the main flow a remote medical device performs a measurement and sends it to an e-health application provider's (AP) application server, which in turn processes the data and notifies the appropriate actors regarding the condition of the patient.

The AP provides an application client to be installed on the device, and the application servers that interact with all the application clients. Both the application client and application server use the data management and communication facilities within the service layer exposed through the service layer APIs.

This flow could be as follows:

- The sensor on the medical device performs a measurement and reports it to the application client on the device.
- The application client (e.g. an e-health application) uses the service layer API to reach the service layer (provided by M2M service provider) within the device to transfer data to the application server. When application level data privacy is required, the application client on the device must encrypt the sensor data before passing the data to the service layer. Since the data must be kept private from service layer function, the encryption keys and engine used by the application client must be kept within a secure environment that is out of reach of the M2M service provider. This may require a set of secure APIs to reach the application's secure environment. It may however be more convenient that these APIs are bundled with the secure APIs used to reach keys/ environment that secures the service layer, so that each application only deals with one set of APIs.
- The service layer (provided by M2M service provider) passes the data from the device to the M2M service provider servers.
- The M2M service layer at the server side passes the data to the e-health application server.
- At this point, the application needs to prepare to notify any interested parties (caregivers) that have subscribed to receive notifications regarding the status or data received about a patient. However, when application data is encrypted and redaction is to applied, more intelligence must be applied regarding who is authorized to receive a notification regarding status update. This may be done as follows:
- After the e-health application server receives the data from M2M SP server, it decrypts the data, analyzes and performs redactions based on application policies (possibly with help of policy servers). This produces multiple versions of the initial data (one at each redaction level). The application server then re-encrypts each redacted version. Each encrypted version needs to be tagged based on the redaction level (RL) it contains and possibly the authorization level (AL) it requires for viewing.
- The application server passes the tagged data (multiple files) to the M2M service provider server (the service layer server)
- The M2M SP server will then sends a notification to each of the subscribers as long as their AL is at or above the level required to view any of the data just received. This means a separate authorization server may have initially performed an authorization of each user that requests to subscribe to data regarding each patient. The authorization would need to assess the identity of the user, her role and the claimed AL before registering the user for notifications. It is possible that the authorization server upon assertion of AL for each user provide the necessary decryption keys for receiving encrypted redacted data to the user's device. In that case, the device that the user is using needs to be authenticated based on a verifiable identity (an identity that is bound to a tamper-proof identity within the secured environment). Alternatively, the decryption keys may be present within the user devices (e.g. specific USB stick!) through other means. In either case a mechanism must exist to release decryption keys stored with an authenticated device's secure storage based on the user authorization and thus a binding of user and device authentications may be important.

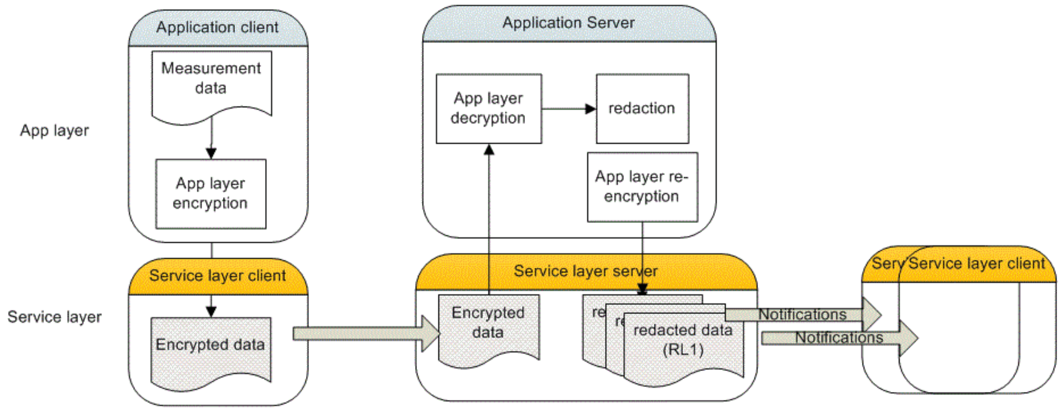


Figure 7-10 Dealing with Redaction in an M2M system separating Application layer and Service layer. The Service layer functions are provided by M2M service provider, while application layer functions are provided by application provider.

7.3.7 Alternative flow

7.3.7.1 Alternative flow No 1

One alternative flow is when a user requests information regarding a patient without having previously subscribed for any notifications. The M2M SP server must first refer the user to the authorization server to assert the user's authorization level (AL) before serving the user with a response.

7.3.7.2 Alternative flow No 2

One alternative flow is when a user requests to provide instruction commands regarding a patient to a remote device. The service must make sure that the user has the proper AL to issue the command.

7.3.7.3 Alternative flow No 3

One alternative flow is when users are categorized not based on authorization levels but based on the level of their responsiveness. For instance, a life-critical event must cause the emergency responders to receive notifications and act very quickly, while a less critical event may only lead to a family member to be alerted. The subscription/ notification system should provide this level of granularity, i.e. information can be tagged based on criticality level. There must also be a policy engine that categorize the data based on its criticality level (CL).

7.3.8 Post-conditions

7.3.8.1 Normal flow

Multiple versions of patient record exist for multiple redaction levels at the M2M service provider servers. Each user can pull the version corresponding to her AL after she has been notified about presence of new data. The server can serve the data based on its RL tagging or AL tagging.

7.3.8.2 Alternative flow No 3

Data is tagged with criticality level and served to each user according to their level of responsiveness.

7.3.9 High Level Illustration

Not provided

7.3.10 Potential requirements

1. The M2M system shall support M2M applications with establishing a security context for protecting the privacy of application data from the underlying M2M service.

This means support of synchronous exchanges required by identification/ authentication/ or other security algorithms for establishment of security associations (keys, parameters, algorithms) for end-to-end encryption and integrity protection of data. Furthermore, any exchanges for establishing the M2M application security context can use the security context at underlying layers (e.g. M2M service layer) to protect the exchanges (as another layer of security), but the M2M application security context, once established, would be invisible to the M2M system.

2. The M2M system must support mechanisms for binding identities used at service layer and/or application layer to the tamper proof identities that are available within the device secured Environment.

Anchoring higher layer identities to a low level identity (e.g. identities that are protected at the hardware or firmware level) is needed to be able to securely verify claimed identities during device authentication processes at various levels. Also APIs providing lower layer identities to application layer for the purpose of binding application layer identities and lower layer identities.

3. M2M devices and M2M system shall support provisioning of application specific parameters and credentials prior and/or after field deployment, while preserving the privacy of provisioned material from M2M system if needed.

This means the M2M devices must support identities and credentials that are independent of the M2M system provider credentials and could be used for delivery of application specific parameters/credentials.

4. When M2M application data security is independent of M2M system, the Secured Environment within devices or infrastructure entities shall provide separation between the secured environments for each application and the secured environment for M2M service layer.
5. The secure environment described in requirement above shall provide both secure storage (for keys, sensitive material) and secure execution engine (for algorithms and protocols) for security functions for each application or service layer.
6. The security functions provided by the Secured Environment should be exposed to both M2M service layer and M2M applications through a set of common APIs that allow use of Secured Environment of each of M2M service layer and M2M applications in a uniform fashion.
7. The M2M service layer must be able to perform authorization before serving users with sensitive data.
8. The authorization process should support more than two authorization levels and the service layer must be able to accommodate response/ notifications to the users based on their level of authorization.
9. The M2M service layer must accommodate tagging of opaque application data for various purposes, such as urgency levels, authorization/redaction levels, etc.
10. There must be a mechanism to allow the M2M application or service layer to bind user credentials/ authorizations to device credentials, such that credentials within the device can be used for security purposes during or after a user is authenticated/ authorized.
11. The M2M service layer must be able to accommodate delay requirements for the application based on the tagging applied to the application data. For instance, data that is marked critical must create notifications for first-level responders.
12. Any software client, especially those performing security functions (e.g. authentication clients) must be integrity protected (signed) and verified after device power up/reset or before launch. Widely deployed standards such PKCS#7 or CMS should be used for code signing.

8 Public Services Use Cases

8.1 Street Light Automation

8.1.1 Description

Street Light Automation can be considered as part of the City Automation (ETSI classifier) vertical industry segment – and related to others e.g. Energy, Intelligent Transportation Systems, etc.

Industry segment organisations: none known

Industry segment standards: none known

Deployed: with varying functionality, in multiple countries

Street Light Automation Goals

- Improve public safety
- Reduced energy consumption / CO2 emissions
- Reduce maintenance activity

Methods

- Sensing and control
- Communications
- Analytics

A street light automation service provider, provides services to control the luminosity of each street light dependent upon (resulting in 10 sub-use cases):

- **Local (street level)**

1. Light sensors
2. Power quality sensors
3. Proximity sensors (civilian or emergency vehicles, pedestrians)

- **Street light automation service provider operation center**

4. Policies (regulatory & contractual)
5. Ambient light analytics (sunrise/sunset, weather, moonlight, etc.)
6. Predictive analytics (lights parts of streets predicted to be used, etc.)

- **Communications received from other service providers**

7. Traffic light service (emergency vehicle priority)
8. Emergency services (vehicle routing, police action, etc.)
9. Road maintenance service (closures and/or diversions)
10. Electricity service (power overload)

8.1.2 Source

1. Cisco Systems – from public document research

2. “Street Light Control” use case identified in [i.5] ETSI TR 102 897

8.1.3 Actors

1. Street light automation application service provider, has the aim is to adjust street light luminosity.

2. Street light devices have the aim is to sense, report, execute local and remote policies, illuminate street.

3. Traffic light application service provider, has the aim is to enhance their emergency vehicle service using street lighting.

4. Emergency services application services provider, have the aim is to brightly illuminate police action areas and brightly illuminate planned path of emergency vehicles.

5. Road maintenance application service provider, has the aim is to obtain extra street light signaling near closed roads.

6. Electricity application service provider, has the aim is to have electricity consumers reduce their load when an overload is declared.

8.1.4 Pre-conditions

See sub-case flows.

8.1.5 Triggers

See sub-case flows.

8.1.6 Normal Flow

1. Sub use case 1 - Local: Light sensors

Summary: (no atomic action steps)

Trigger: Detected light level moves below/above threshold

Action: Increase/decrease luminosity in a set of street lights

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Street lights” message the Street light system that street light sensors have detected light level movement below/above threshold.

2. Street light system informs the “street light operation centre” with the street light sensor information.

3. “Street light operation centre” messages the Street light system with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.

4. Street light system messages the “street lights” with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.

5. Optionally (normal case), if “street lights” receive a control command from the Street light system within some time, then, “street lights” increase/decrease luminosity in a set of street lights according to “street light operation centre” policy.

6. Optionally (alternative case), if “street lights” do not receive a control command from the Street light system within some time, then, “street lights” increase/decrease luminosity in a set of street lights, according to local policy.

Note that the terminology “policy” refers to a set of rules which may be dependent upon variables output from analytics algorithms.

2. Sub use case 2 - Local: Light sensors

Local: Power quality sensors

Summary: (no atomic action steps)

Trigger: Detected input voltage level moves above/below threshold

Action 1: Send alert message to electricity service provider

Action 2: Decrease/increase energy applied to a set of street lights

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Street lights” message the Street light system that street light power sensors have detected input voltage level movement above/below threshold
2. Street light system informs the “street light operation centre” with the street light sensor information
3. “Street light operation centre” messages the Street light system with an alert message to “electricity service provider” according to “street light operation centre” policy.
4. Street light system informs “electricity service provider” of alert message.
5. “Street light operation centre” messages the Street light system with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.
6. Optionally (normal case), if “street lights” receive a control command from the Street light system within some time, then, “street lights” increase/decrease luminosity in a set of street lights according to “street light operation centre” policy.
7. Optionally (alternative case), if “street lights” do not receive a control command from the Street light system within some time, then, “street lights” increase/decrease luminosity in a set of street lights, according to local policy

3. Sub use case 3 - Local: proximity sensors (civilian or emergency vehicles, pedestrians)

Summary: (no atomic action steps)

Trigger: Civilian or emergency vehicle or pedestrian detected entering/leaving street section

Action: Increase/decrease luminosity in a set of street lights

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Street lights” message the Street light system that street light power sensors have detected civilian or emergency vehicle or pedestrian detected entering/leaving street section.
2. Street light system informs the “street light operation centre” with the street light sensor information.
3. “Street light operation centre” messages the Street light system with a control message to increase/decrease luminosity according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.
5. Optionally (normal case), if “street lights” receive a control command from the Street light system within some time, then “street lights” increase/decrease luminosity in a set of street lights according to “street light operation centre” policy.

6. Optionally (alternative case), if “street lights” do not receive a control command from the Street light system within some time, then, “street lights” increase/decrease luminosity in a set of street lights, according to local policy.

4. Sub use case 4 – Operation Centre: Policies (regulatory & contractual)

Summary: (no atomic action steps)

Trigger: SLA non-conformity for low intensity imminent

Action: Increase luminosity in a set of street lights to keep within SLA

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. The “street light operation centre” detects through analytics that an SLA regarding minimum street light intensity is in danger of not being met.

2. “Street light operation centre” messages the Street light system with a control message to increase luminosity according to “street light operation centre” policy.

3. Street light system messages the “street lights” with a street light control message to increase luminosity according to “street light operation centre” policy.

5. Sub use case 5 - Operation centre: Ambient light analytics (sunrise/sunset, weather, moonlight)

Summary: (no atomic action steps)

Trigger 5a: A band of rain moves across an area of street lights

Action 5a: Increase/decrease luminosity in a rolling set of street lights

Trigger 5b: Sunrise/sunset is predicted to occur area in 30 minutes

Action 5b: Decrease/increase luminosity in a rolling set of street lights

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. The “street light operation centre” detects through analytics that (5a) a band of rain is moving across an area of street lights, or (5b) Sunrise/sunset is predicted to occur area in 30 minutes.

2. “Street light operation centre” messages the Street light system with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.

3. The Street light system messages the “street lights” to increase/decrease luminosity in a set of street lights according to “street light operation centre” policy.

6. Sub use case 6 - Operation centre: Predictive analytics (lights parts of streets predicted to be used)

Summary: (no atomic action steps)

Precondition: Vehicle paths are tracked via proximity sensors and a route model is generated

Trigger: A vehicle enters a street section which has 85% probability of taking the next left turn

Action: Increase luminosity on current street section ahead and also on street on next left

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Street lights” message the Street light system that street light power sensors have detected civilian or emergency vehicle entering street section
2. Street light system informs the “street light operation centre” with the street light sensor information
3. “Street light operation centre” messages the Street light system with a control message to increase/decrease luminosity according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to increase/decrease luminosity according to “street light operation centre” policy.

7. Sub use case 7 - From other service providers: Traffic light service input (emergency vehicle priority)

Summary: (no atomic action steps)

Trigger: An emergency vehicle is approaching a junction

Action: Increase luminosity in street lights along streets leading away from junction

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Traffic light service provider” messages the Street light system that emergency vehicle approaching street junction from certain direction.
2. Street light system informs the “street light operation centre” with the street junction information.
3. “Street light operation centre” messages the Street light system with a control message to increase luminosity according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to increase luminosity according to “street light operation centre” policy.

8. Sub use case 8 - From other service providers: Emergency services input (vehicle routing, police action)

Summary: (no atomic action steps)

Trigger 8a: An emergency vehicle route becomes active

Action 8a: Increase luminosity in street lights along vehicle route

Trigger 8b: An area is declared as having an active police action

Action 8b: Increase luminosity in street lights within police action area

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Emergency services provider” messages the Street light system that (8a) emergency vehicle street route is active, or (8b) an area is declared as having an active police action
2. Street light system informs the “street light operation centre” with the street junction information
3. “Street light operation centre” messages the Street light system with a control message to increase luminosity according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to increase luminosity according to “street light operation centre” policy.

9. Sub use case 9 - From other service providers: Road maintenance service input (closures and/or diversions)

Summary: (no atomic action steps)

Trigger 9a: A road is closed

Action 9a: Program a changing luminosity pattern in street lights near to closed road

Trigger 9b: A route diversion is activated

Action 9b: Program a changing luminosity pattern in street lights along the streets of the diversion

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Road Maintenance service provider” messages the Street light system that (9a) a road is closed, or (9b) a route diversion is activated
2. Street light system informs the “street light operation centre” with the road maintenance information
3. “Street light operation centre” messages the Street light system with a control message to set lights to changing luminosity pattern according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to set lights to changing luminosity pattern according to “street light operation centre” policy.

10. Sub use case 10 - From other service providers: Electricity service input (power overload)

Summary: (no atomic action steps)

Trigger: A power overload situation is declared

Action: Decrease luminosity in a set of street lights

Detailed flow (no confirmation, etc. – actors in “quotes”, system under study in italics)

1. “Electricity service provider” messages the Street light system that (9a) that an overload condition exists across some area.
2. Street light system informs the “street light operation centre” with the overload condition information
3. “Street light operation centre” messages the Street light system with a control message to decrease luminosity according to “street light operation centre” policy.
4. Street light system messages the “street lights” with a street light control message to decrease luminosity according to “street light operation centre” policy.

8.1.7 Alternative flow

In the case of loss of communications, street lights have local policies which they obey.

8.1.8 Post-conditions

Street light luminosity or luminosity pattern is adjusted as needed.

8.1.9 High Level Illustration

🟩 =
□ =

Figure 8-1 Street Light Automation High Level Illustration

8.1.10 Potential Requirements

Generic (needed by two or more verticals or applications)

1. The M2M solution shall support the ability to collect information from M2M devices.
2. The M2M solution shall support the ability to deliver collected information from M2M devices to M2M applications.
3. The M2M solution shall support control commands (for devices) from M2M applications.
4. The M2M solution shall support control commands for groups of M2M devices.
5. The M2M solution shall support the ability to receive device application software from M2M applications.
6. The M2M solution shall support the ability to deliver device application software to M2M devices.
7. The M2M solution shall provide mechanisms for information sharing, i.e. receiving information from M2M applications (information providing) to be consumed by other M2M applications (information consuming).
8. The M2M solution shall provide charging mechanisms for information sharing among M2M applications.
9. The M2M solution shall support the ability to provide an estimate of the time period from when a device sent a message to the M2M solution until when it responded with a message to the device.
10. The M2M solution shall provide security context (authentication, encryption, integrity protection) for secure connection between entities. The security context shall include mechanisms and techniques on how to setup a security connection , and where the security connection information is stored and how to establish the secure connection
11. The M2M service layer shall provide security mechanisms to facilitate the end to end security of M2M applications.
12. The M2M service layer shall provide security mechanisms to avoid compromising the end to end security of M2M applications.

Specific (to this vertical/use case)

None

Note that the terminology:

- "Device application software" refers to application software that runs on a device including programs, patches, program data, configuration, etc.
- "M2M application" is any application that makes use of the M2M service layer - some form of prior agreement may be needed.

Security Considerations

1. Attack vectors and example impacts:
    - By sending false reports of sensors to applications
    - Energy provider overdriving voltage
2. By sending false control commands to devices
    - Blackout to obscure crime
3. By blocking valid messages

- Energy wastage

8.2 Use Case on Devices, Virtual Devices and Things

8.2.1 Description

The municipality of a Smart City operates an Application Service that monitors traffic flow and switches traffic lights depending on traffic. This “traffic application” controls the traffic lights and a couple of surveillance cameras to observe traffic flow.

The traffic application makes several of the surveillance cameras discoverable in the M2M System and potentially allows access to the data (the video streams) of these cameras. The surveillance cameras can be searched and discovered in the M2M System based on search criteria such as type (e.g. video camera for traffic) and other meta-data (e.g. location or activation state).

In addition to (physical) devices the traffic application publishes “virtual devices” that act similar to sensors and provide derived data such as: number of vehicles that passed during the last minute/hour, average speed of vehicles …

Also these “virtual devices” can be searched and discovered in the M2M System based on type and other meta-data.

However, in contrast to the previous case (real devices) virtual devices only implemented as software and do not require a Connectivity Layer. They are data structures published by the traffic application.

The traffic application charges other applications to receive data from these virtual devices.

Finally, the traffic application also publishes “things” in the M2M System like roads and intersections. Other “things” the traffic application might publish are phased traffic lights (green wave).

“Things” are similar to “virtual devices” but have relations to other “things” (e.g. a section of a road lies between two intersections).

A “street”, published by the traffic application, provides information on the average speed of traffic, congestion level, etc. A “series of phased traffic lights” provides information about which traffic lights are in phase, the current minimal/maximal/optimal speed, etc.

The “traffic application” of the Smart City charges other applications to access data from its published “things”.

A second Application Service, a “logistics application” is operated by a company that manages a fleet of trucks to deliver goods all over the country. This “logistics application” provides an optimal route for each truck at any time.

One of the trucks is currently driving in the Smart City. The logistics application has a service level agreement with the traffic application of the Smart City.

The logistics application discovers all things (streets, intersections…) that are relevant to calculate an optimal route for the truck, based on type and location. It uses the published data and is charged for the access to these data.

8.2.2 Source

NEC

8.2.3 Actors

- The municipality of a Smart City (Application Service Provider)
- The fleet management company (Application Service Provider

- The M2M

8.2.4 Pre-conditions

- The municipality of a Smart City operates a “traffic application” that monitors traffic flow and switches traffic lights.
- The fleet management company operates a “logistics application” that manages a fleet of trucks.
- Both Applications are using the same M2M Service Capabilities Network (MSCN) operated by the M2M Service provider.
- The traffic application allows the logistics application to access some of its Devices, Virtual devices and Things.

8.2.5 Triggers

None

8.2.6 Normal Flow

- The traffic application creates Virtual devices (e.g. traffic sensors) and Things (e.g. streets, series of phased traffic lights…) for use by other M2M applications in the MSCN of the M2M Service operator.
- The traffic application publishes the semantic description (types, relations, and meta-data) of its Devices (e.g. cameras), Virtual devices and Things in the MSCN of the M2M Service operator. The traffic application restricts discoverability of its Virtual devices and Things to applications provided by business partners of the municipality of a Smart City.
- The traffic application enables access to the data of some of its traffic cameras to all M2M applications, but access to the data of virtual devices and things is restricted to applications of business partners (e.g. the logistics application).
- The logistics application searches the MSCN of the M2M Service operator for things and virtual devices in the vicinity of the truck. Based on the semantic search criteria (described by reference to a taxonomy or ontology) only the things and virtual devices that are useful for calculating the route of the truck are discovered.
- The logistics application reads the data from relevant things and virtual devices and calculates the optimal route for the truck.
- The logistics application is charged by the MSCN of the M2M Service operator for reading the data from things and virtual devices of the traffic application.
- The traffic application is reimbursed for usage of its things and virtual devices.

8.2.7 Alternative flow

None

8.2.8 Post-conditions

None

8.2.9 High Level Illustration

None

8.2.10 Potential Requirements

- The M2M System shall provide a capability to an Application shall be able to create Virtual Devices and Things in the M2M Service Capability Network.

- The M2M System shall provide a capability to an Application shall be able to publish semantic descriptions and meta-data (e.g. location) of its Devices, Virtual Devices and Things in the M2M Service Capability Network.

- The M2M System shall provide a capability to an Application to search for and discover Devices, Virtual Devices and Things in the M2M Service Capability Network based on their semantic descriptions and meta-data. The supported formats of semantic descriptions shall be described in the oneM2M standard.

- The M2M System shall provide a capability to an Application shall be able to control, via the M2M Service Capability Network, access to semantic descriptions and meta-data of its Devices, Virtual Devices and Things.

- The M2M System shall provide a capability to an Application shall be able to allow, via the M2M Service Capability Network, access to its Devices, Virtual Devices and Things to individual other applications.

8.3 Car/Bicycle Sharing Services

8.3.1 Description

As seen clearly, automation already penetrates all aspects of life even in our urban life. The goal of this use case is to describe several automation services which are occurred in different urban space in different life style, bicycle/car sharing services.

Brief Features of Services

- Car Sharing Service

  Car Sharing is to offer a new service model for automobile transportation. Simply, Car Sharing is a self-service, on-demand alternative to car ownership; a service that is offered to urban residents (B2C) and businesses (B2B).

  This service is mainly designed around a particular user profile – first of all, people who live in cities but do not drive a car every day and secondly tourists who live in cities but do not own a car. Thus, people who need a car at short notice but take an alternative to car ownership.

  The brief procedure of this service is 1) joining the membership, 2) unlocking the car door, 3) driving away, 4) parking to any reserved spot provided by the service provider and/or public, and 5) paying as you drive (including gas, insurance, and etc.).

- Bicycle Sharing Service

  Bicycle sharing service is also a new service in which bicycle are made available for shared use to individuals who do not own a bicycle. Generally, bicycle sharing service is run by government agencies.

  The procedure of this service is similar to the car sharing service, but the different type of services such as healthcare service can be combined.

8.3.2 Source

LG Electronics

8.3.3 Actors

- **User**

A user who takes the ownership of the shared things which are car and bicycle.

- **Sensors (or Sensor Devices)**

Sensor Devices can be various based on its usage, and do not have any direct communication interfaces to the M2M Service Platform.

For Car Sharing Service – Door Control Sensor, Tire Pressure Sensor, Fuel Indication Sensor, GPS.

For Bicycle Sharing Service – Lock Control Sensor, Accelerometer, Tire Pressure Sensor, Heart-rate Sensor.

- **Smartphone**

A device which is an intermediate entity and is available to connect from sensors to a M2M Service Platform. The basic role is similar to the general M2M gateway, but it has some sensors and some applications (navigation) itself used by services.

- **M2M Service Platform**

In charge of providing common functionalities for the M2M services. It is mainly in charge of collecting the status and configuration information of sensors and controlling them via the smartphone and/or M2M gateway.

- **M2M Service Providers**

Companies which provide its own M2M services for the user through the M2M Service Platform. The M2M Service Providers can be various according to the types of services.

The providers include Car Sharing Service Provider, Insurance Company, Gas Station, Bicycle Sharing Service Provider, and Healthcare Service Provider.

8.3.4 Pre-conditions

See sub-case flows.

8.3.5 Triggers

See sub-case flows.

8.3.6 Normal Flow

1. Sub use case 1 – Car Sharing Case

- **Trigger**:

A user wants to take an ownership of the car.

- **Pre-conditions:**

The user preliminary joins a membership of the Car Sharing Service.

Sensors built in the car are required to periodically (normal) and non-periodically (urgent) send sensor data to the M2M Service Platform based on the trigger defined by the M2M Service Providers.

The M2M Service Platform collects and manages data and configurations related to the services. Generally, each service has its own data and configuration set, simply called resources.

The M2M Service Providers in the service domain have a service agreement each other for unified services.

The Smartphone has a navigation and car sharing application.

- **Detailed Flow Descriptions:**

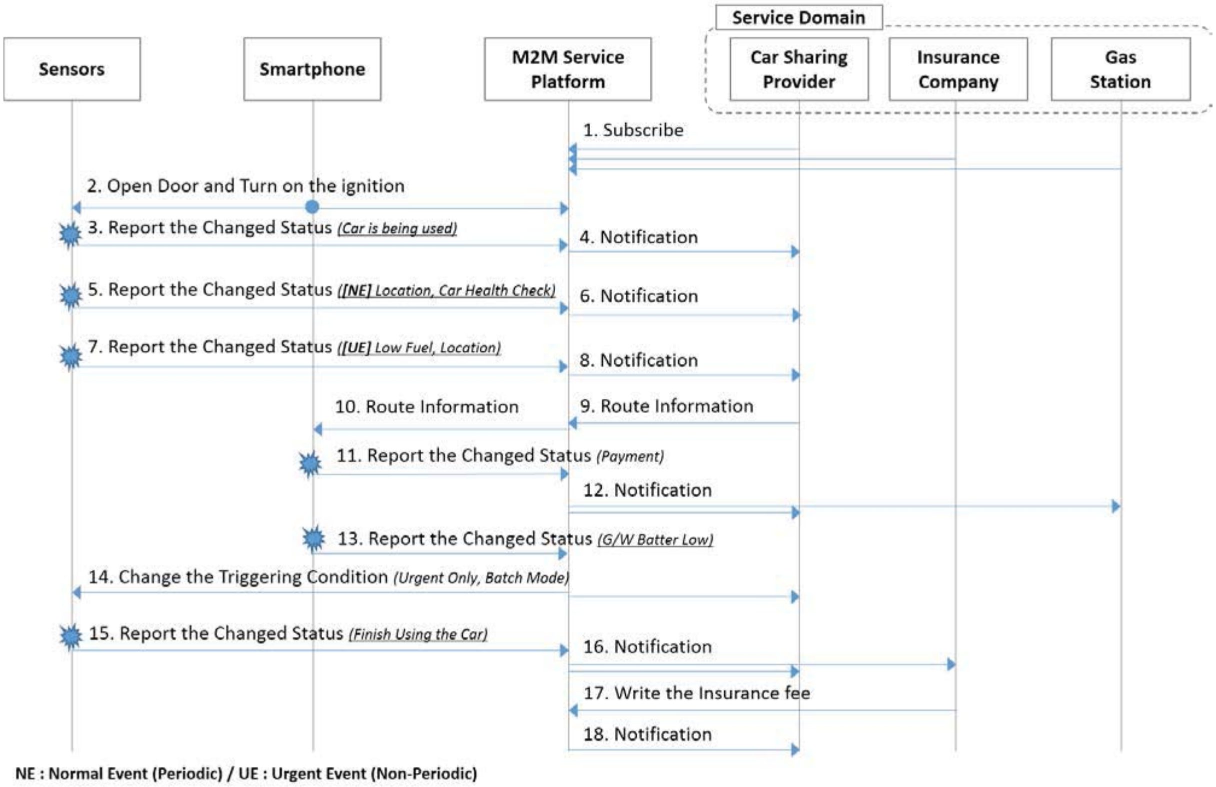

Service Domain
Sensors
Smartphone
M2M Service Platform
Car Sharing Provider
Insurance Company
Gas Station
1. Subscribe
2. Open Door and Turn on the ignition
3. Report the Changed Status (Car is being used)
4. Notification
5. Report the Changed Status ([NE] Location, Car Health Check)
6. Notification
7. Report the Changed Status ([UE] Low Fuel, Location)
8. Notification
10. Route Information
9. Route Information
11. Report the Changed Status (Payment)
12. Notification
13. Report the Changed Status (G/W Batter Low)
14. Change the Triggering Condition (Urgent Only, Batch Mode)
15. Report the Changed Status (Finish Using the Car)
16. Notification
17. Write the Insurance fee
18. Notification
NE : Normal Event (Periodic) / UE : Urgent Event (Non-Periodic)

Figure 8-2 Car Sharing Normal Flow

1. The Applications of each Service Provider in the service domain register and subscribe to changes of resources (or information) about the Car Sharing Service in the M2M Service Platform.

2. Since each resource in the M2M Service Platform is owned by the Car Sharing Provider, Insurance Company and Gas Station, if an application needs to access another resource, it shall request proper access right of the resources and grant that request if appropriate and based on the service agreement.

3. As the user finds a shared car, opens the car door and turns on the ignition using interfaces of the Smartphone such as Bluetooth and NFC, if the user is authorized.

4. The Sensors report the changed status to the M2M Service Platform via the Smartphone as a gateway when the specific condition is triggered. (Car is just being used)

5. The M2M Service Platform notifies the Car Sharing Service Provider of the changed status. (Note: The Car Sharing Service Provider can update the situation that the car is being used on its website)

6. (Normal Reporting Case for managing the Service) The Sensors report the changed status to the M2M Service Platform via the Smartphone when the specific condition is triggered. (Periodic location reporting and car health check for maintenance reasons)

7. The M2M Service Platform notifies the Car Sharing Service Provider of the changed status. (Note: Agreement on privacy policy of location is preliminary confirmed)

8. (Urgent Reporting Case for handling any emergency) The Sensors report the changed status to the M2M Service Platform via the smartphone as a gateway when the specific condition is triggered. (The fuel is low)

9. The M2M Service Platform immediately notifies the Car Sharing Service Provider of the changed status.

10. The Car Sharing Service Provider finds out the nearest Gas Station according to the received location information and a service agreement between the Car Sharing Service Provider and the Gas Station, and the Provider sends the route information to M2M Service Platform.

11. The M2M Service Platform notifies the Smartphone of the route information.

12. After filling the fuel, the user virtually pays the fuel fee by using the Smartphone's NFC tag. The payment information is reported to the M2M Service Platform.

13. The M2M Service Platform notifies the Car Sharing Provider and the Gas Station of the payment information. (Note: This procedure is for the Car Sharing Provider to pay Gas Station the fuel fee instead of the user)

14. Afterwards, due to the low battery of the Smartphone (less than 30% remain), the Smartphone reports the changed status to the M2M Service Platform.

15. The M2M Service Platform automatically changes the subscription and reporting attributes of the Sensors and the Car Sharing Service Provider. (For example, if the Platform changes the subscription attributes to "only emergency case", only emergency subscription case will be notified. The others cannot be notified, but at the end of service, batch-mode)

16. As the user arrives at the destination, and turns off the ignition, the sensors report the accumulated information, normal event subscription information, to the M2M Service Platform via smartphone.

17. The M2M Service Platform notifies the Car Sharing Provides and Insurance Company of the usage of the shared car.

18. The Insurance Company stores the insurance fee by writing onto the Car Sharing Service Provider's resource in the M2M Service Platform, in this case the Insurance Company preliminary acquires proper access right to write.

19. The M2M Service Platform notifies the Car Sharing Provides of the insurance fee.

- **Post-conditions:**

  - The User will pay as him/her drive according to the recorded data.

  - The Car Sharing Service Provider can update the position and status of the car on its website using the recorded data. Thus, next users can make use of the Car Sharing Service.

2. Sub Use Case 2 – Bicycle Sharing Service

- **Trigger:**

  A user wants to take an ownership of the bicycle.

- **Pre-conditions:**

  The user preliminary joins a membership of the Bicycle Sharing Service.

  The sensors built in the car and in the smartphone are required to periodically (normal) and non-periodically (urgent) send sensor data to the M2M Service Platform based on the trigger defined by the Service Provider.

  The M2M Service Platform collects and manages data and configurations related to the services. Generally, each service has its own data and configuration set, simply called resources.

  The Smartphone has a navigation and bicycle sharing application.

  The M2M Service Providers in the service domain have a service agreement each other for unified services.

- **Detailed Flow Descriptions:**

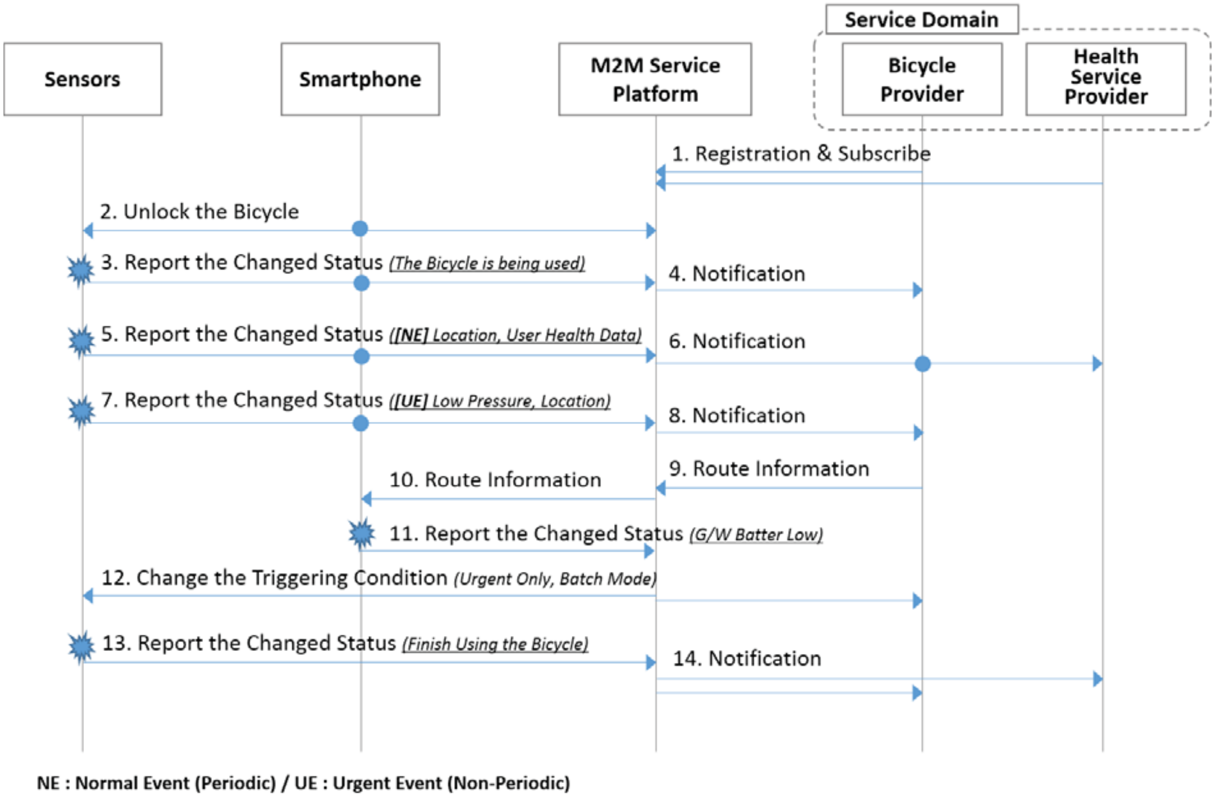

Service Domain
Sensors
Smartphone
M2M Service Platform
Bicycle Provider
Health Service Provider
1. Registration & Subscribe
2. Unlock the Bicycle
3. Report the Changed Status (The Bicycle is being used)
4. Notification
5. Report the Changed Status ([NE] Location, User Health Data)
6. Notification
7. Report the Changed Status ([UE] Low Pressure, Location)
8. Notification
9. Route Information
10. Route Information
11. Report the Changed Status (G/W Batter Low)
12. Change the Triggering Condition (Urgent Only, Batch Mode)
13. Report the Changed Status (Finish Using the Bicycle)
14. Notification
NE : Normal Event (Periodic) / UE : Urgent Event (Non-Periodic)

Figure 8-3 Bicycle Sharing Normal Flow

1. The Applications in the service domain register the service and subscribe to changes of information about the Bicycle Sharing Service.

2. Since each resource in the M2M Service Platform is owned by the Bicycle Sharing Service Provider and Health Service Provider, if an Application needs to access another resource, it shall request proper access right of the resources and grant that request if appropriate and based on the service agreement.

3. To unlock the bicycle, the user tags the locker of the bicycle through the NFC interface.

4. The Sensors report the changed status to the M2M Service Platform via the smartphone as a gateway when the specific condition is triggered (for example, the bicycle is being used).

5. The M2M Service Platform notifies the Bike Sharing Service Provider of the changed status. (Note: The Bicycle Sharing Service Provider can record the situation on its web-site that the car is being used).

6. (Normal Reporting Case for managing the Service) The heart-rate of the user is continuously collected by the heart-rate sensor on the handlebar, and the health-related information such as heart-rate, location, time is reported periodically to the Service Operator.

7. The M2M Service Platform notifies the Bicycle Sharing Service Provider and the Health Service Provider of the health Service information.

8. (Urgent Reporting Case for handling any emergency) While riding the bicycle, the tire pressure sensor detects the low pressure of the front tire, the information is immediately sent to the M2M Service Platform via the Smartphone with location information.

9. The M2M Service Platform notifies the Bicycle Sharing Service Provider of the changed status.

10. The Bicycle Sharing Service Provider finds out the nearest bike repair shop according to the received location information, and the Provider sends the route information to M2M Service Platform.